TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Septimia Sarbu

Information Theoretic Analysis of the Structure-Dynamics Relationships in Complex Biological Systems

Julkaisu 1382 • Publication 1382

Tampere 2016

Tampereen teknillinen yliopisto. Julkaisu 1382
Tampere University of Technology. Publication 1382

Septimia Sarbu

Information Theoretic Analysis of the Structure-Dynamics Relationships in Complex Biological Systems

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Tietotalo Building, Auditorium TB222, at Tampere University of Technology, on the 20th of May 2016, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2016

ISBN 978-952-15-3734-9 (printed)
ISBN 978-952-15-3738-7 (PDF)
ISSN 1459-2045

Information theoretic analysis of the structure-dynamics relationships in complex biological systems

Septimia Sarbu
Department of Signal Processing
Tampere University of Technology

March 16, 2016

Abstract

Complex systems and networks is an emerging scientific field, with applications in every area of human enquiry, for which a solid theoretical, computational and experimental foundation is lacking. As our technological capability of generating and gathering vast amounts of data from such systems is increasing, precise methods are needed to describe, analyse and synthesize such systems. Systems biology is a prime example of an interdisciplinary field aiming at tackling the complexity of biological organisms and dedicated to understanding their organizing principles and to devising efficient intervention strategies for curing diseases.

A very important topic in the study of complex systems and networks is to uncover the laws that govern their structure-dynamics relationships. A complete description of the system's behaviour as a whole can only be achieved if the structure and the dynamics are investigated together, as well as the intricate ways in which they influence each other. The understanding of structure-dynamics relationships is a key step in the control of complex systems and networks. For example, in biology, understanding these relationships in organisms would enable us to find more precise drug targets and to design better drugs to cure diseases. In gene regulatory networks, it would help devise control strategies to change the network from faulty states that correspond to disease states, to normal states that give the healthy phenotype. When we observe a dynamical behaviour that is different from the normal, healthy one, the knowledge about the structure-dynamics relationships would help us identify which part of the structure gives rise to such behaviour. Then, we would know where and how to change the structure, to return the system to its normal dynamics, that is, to obtain a desired dynamical behaviour.

A feasible way of investigating the structure-dynamics relationships is by measuring the amount of information that is communicated in the system and by analysing the patterns of information propagation within its elements. These objectives can

be achieved by means of information theory. To this end, with concepts from Kolmogorov complexity and from Shannon's information theory, we create novel analysis methods of the structure-dynamics relationships in two models of complex systems: an executable model of the human immune systems and the random Boolean network model of gene regulatory networks.

In these endeavours, the information-theoretic means of identifying and measuring the information propagation in complex systems and networks needs to be improved and extended. Research is needed into the theoretical foundations of information theory, to refine existing equations and to introduce new ones that can give more accurate results in the investigation of the propagation of information and its applications to the structure-dynamics relationships. To this end, we bring analytical contributions to the generalization of Shannon's information theory, named Rényi's information theory. Thus, we continue the development of the theoretical foundations of information theory, for new and better applications in complex systems science and engineering.

The goal of this thesis is to characterize various aspects of the structure-dynamics relationships in models of complex biological systems, by means of information theory. Moreover, our goal is to prove that information theory is a model independent analysis framework that can be applied to any class of models. We pursue our objective, by analysing two different classes of models: an executable model of the human immune system and the random Boolean network model of gene regulatory networks.

In the executable model of the regulation of cytokines within the human immune system, our aim is to develop computationally feasible analysis methods that can extract meaningful biological information from the complex encoding of the dynamical behaviour of different perturbations of the wild type system. We aim at classifying several structural perturbations of the system, using only their dynamical information. We endeavour to create methods that can make predictions about the structural parameters that should be changed in order to obtain a desired dynamical behaviour. These conclusions have direct applications to the fine-tuning of the real-world biological experiments performed on the system, of whose computational model we analyse. The benefits of our predictions would be increased efficiency and increased reduction of the time required to optimize the parameters of the real-world biological experiments.

In the random Boolean network model of gene regulatory networks, our goal is to develop an experimental order parameter that can characterize the dynamical

regime of the network, from the dynamical behaviour that simulates that obtained from the measurements of real-world biological experiments. Moreover, we aim at proving that structural information is hidden in the dynamics of random Boolean networks and that it can be extracted with methods from information theory. We study ensembles of random Boolean networks from two distinct structural classes, which take into account the stochasticity present in real biological systems.

Another goal of this study is to bring analytical contributions to the field of Rényi's information theory, which is a generalization of Shannon's information theory. Recently, it has found novel applications in the study of the structure-dynamics relationships in complex systems and networks.

Acknowledgements

The research work presented in this thesis was conducted between 2011 – 2015, in professor Olli Yli-Harja's group, Computational Systems Biology, at the Department of Signal Processing, at Tampere University of Technology, in Finland and in professor Ilya Shmulevich's group, at the Institute of Systems Biology (ISB), in Seattle, USA.

I would like to express my deepest gratitude to professor Olli Yli-Harja, for giving me the great opportunity of working in the Computational Systems Biology group, for all his guidance and advice I received in pursuing my postgraduate degree. I would like to thank professor Matti Nyker, and Dr.Tech. Juha Kesseli, from University of Tampere, for our close collaboration on the work presented in this thesis.

I would like to thank the Graduate School in Electronics, Telecommunications and Automation, GETA, for the scholarship received for my research and to Marja Leppäharju, for all her support with the administrative matters. I wish to acknowledge the help received from Ulla Siltaloppi and Virve Larmila, from the Department of Signal Processing, for making the administrative matters easy and straightforward.

I am extremely grateful to professor Ilya Shmulevich, for the opportunity of a research exchange visit to ISB, for all his guidance and collaboration, which helped me further develop my skills as a researcher. I would like to thank everyone I met and interacted with at ISB for an extremely rewarding research visit, both professionally and personally. I would especially like to thank Avital Sadot for our collaboration, which started before my visit to ISB.

I wish to thank the colleagues and friends from the Department of Signal Processing, in Tampere University of Technology, for all the relaxing coffee breaks. I would like to thank my Romanian and international friends here in Tampere, for all the wonderful and joyful moments we spent together during these summers and winters.

I would like to express my entire gratitude to my family, for all the love, support and encouragements they have given me throughout the years, in pursuit of my dreams and my career. Thank you for always being there for me.

TABLE OF CONTENTS

Mathematical notations

The following mathematical notations pertain to the chapter titled "Information theory". This section of notations refer to the sections "Shannon's information theory" and "Generalizations of Shannon's information theory":

$\mathbb{R}$	the collection/set of real numbers
X	a random variable
x	the value taken by the random variable X
$\mathcal{E}_X$	the ensemble of X
$p_X(x)$	the probability mass function, if X is a discrete random variable or the probability density function, if X is a continuous random variable
$\mathrm{H}(X)$	the entropy of X
$\mathbb{E}X$	the expectation of X
$\mathrm{Var}(X)$	the variance of X
$X\|Y = y$	the conditional random variable X, given that the random variable Y takes the value y
$p_{X\|Y}(x\|y)$	the conditional probability mass function
$\mathrm{H}(X\|Y)$	the conditional entropy
p, q	probability mass functions
$\mathrm{D}_{\mathrm{KL}}(p\|\|q)$	the Kullback-Leibler divergence between the probability mass functions p and q
$\mathrm{D}^*_{\mathrm{KL}}(p(X\|Y)\|\|q(X\|Y))$	the conditional Kullback-Leibler divergence between the conditional probability mass functions p and q

$\mathbb{P}_p(X = x)$	the probability that the random variable X takes the value x, with respect to the probability mass function p
$f : [a\ \ b] \to \mathbb{R}$	a continuous function that takes value from an interval and produces real values
$\mathcal{P}$	a collection of probability values
H_α	Rényi's entropy of order α
$\mathrm{D}_\alpha(p \parallel q)$	Rényi's α-divergence between the probability mass functions p and q
$\mathrm{D}^*{}_\alpha(p(X\|Y)\|\|q(X\|Y))$	the conditional Rényi's α-divergence between the conditional probability mass functions p and q
$\mathcal{U}(0,1)$	the uniform probability distribution on the interval $(0,1)$

The following notations refer to the section "Kolmogorov complexity":

x	a string
$\mathrm{K}(x)$	the Kolmogorov complexity of the string x
$x*$	a binary program that computes x
$\mathrm{K}(x\|y)$	the conditional Kolmogorov complexity of the string x, given the string y
$\mathrm{E}(x,y)$	the information distance between x and y
$\mathrm{NID}(x,y)$	the normalized information distance between x and y
C_x	the size of the compressed string x
$\mathrm{NCD}(x,y)$	the normalized compression distance between x and y

The following mathematical notations pertain to the chapter titled "Multidimensional scaling":

N	the total number of data points
$\mathbf{X}$	the points in the low-dimensional Euclidean space
$\mathbf{D}$	the matrix of Euclidean distances between the elements of $\mathbf{X}$
$\mathbf{P}$	the matrix of proximities in the high-dimensional space
$\hat{\mathbf{D}}$	the matrix of distances in the low dimensional Euclidean space, which approximates $\mathbf{D}$
σ_r	the raw stress criterion
σ	Kruskal's Stress-1 criterion

The following mathematical notations pertain to the chapter titled "Random Boolean networks":

N	the total number of nodes
CM	the connectivity matrix of the network
N_i	the total number of in-coming edges
N_o	the total number of out-going edges
$\overline{N_i}$	the mean number of in-coming edges
$\overline{N_o}$	the mean number of out-going edges
O_i	the i^{th} node of the network, $\forall i = 1, \ldots, N$
N_{O_i}	the number of neighbours of the node O_i, $\forall i = 1, \ldots, N$
$N_{i_{O_i}}$	the number of input nodes to the node O_i, $\forall i = 1, \ldots, N$
$N_{o_{O_i}}$	the number of output nodes from the node O_i, $\forall i = 1, \ldots, N$
K, K_{in}	the fixed in-degree of a node
K_{out}	the fixed out-degree of a node
$\overline{K_{in}}$	the mean in-degree of a node
$\overline{K_{out}}$	the mean out-degree of a node
O_i^k	one of the neighbour nodes of the node O_i, $\forall k = 1, \ldots, N_{O_i}$
t	the time index
$\mathrm{S_{RBN}}$	the state of the entire network
f	a Boolean function of n variables
p	the Boolean function bias
s	the expectation of the average sensitivity
N_s	the number of time steps of one trajectory, excluding the initial state
N_n	the number of times the network is restarted in a random initial state
TS	the matrix of trajectories
CC_i	the local directed clustering coefficient of the node O_i, $\forall i = 1, \ldots, N$
CC	the average directed clustering coefficient of the entire network.

Abbreviations

MI	the mutual information
nMI	the normalized mutual information
RMI	the Rényi mutual information
CMI	the conditional mutual information
CRMI	the conditional Rényi mutual information
TE	the transfer entropy
RTE	the Rényi transfer entropy
PMI	the partial mutual information
PRMI	the partial Rényi mutual information
PTE	the partial transfer entropy
PRTE	the partial Rényi transfer entropy
NID	the normalized information distance
NCD	the normalized compression distance
E	the information distance
K	the Kolmogorov complexity
C_x	the length of the compressed string x
HSP-60	the heat shock protein 60
Tregs	the regulatory T cells
nTh	the naïve T cells
$CD4^+CD25^+$ T cells	the regulatory T cells
$CD4^+CD25^-$ T cells	the naïve T cells

CD8 T cells	the cytotoxic T cells
CD4 T cells	the helper T cells, classified into Th1 and Th2 T cells
TLR	the Toll-like receptor
NF-κB	a transcription factor
AKT	a protein kinase(enzyme)
Pyk2	a protein kinase(enzyme)
p38	a protein kinase(enzyme)
ERK	extracellular-regulated kinase
CTLA-4	the cytokine named Cytotoxic T-Lymphocyte Antigen-4
IL-10	the cytokine named Interleukin-10
TGF-β	the transforming growth factor-β
TCR	the T-cell antigen receptor
T-bet	a transcription factor
IFN-γ	the cytokine named Interferon-γ
TNF-α	the cytokine named Tumor Necrosis Factor- α
CD3	a protein complex of the TCR
aCD3	the antibody anti-CD3
GemCell	the software program named Generic Executable Modeling of Cells
MDS	the multidimensional scaling algorithm
nonmetric MDS	the nonmetric multidimensional scaling algorithm
RBN	the random Boolean network
SVM	the support vector machine classification algorithm

List of Figures

Structure of the thesis

The monograph is organized into six chapters. In the first chapter, we present an introduction to the field of complex systems and networks, highlighting their most important properties and the challenges researchers face in the analysis and synthesis of such systems. We emphasize that systems biology is a prime example of a field of complex systems and networks, where the applications of drug discovery and the curing of diseases are of vital importance. The analysis of the structure-dynamics relationships is one of the fundamental questions investigated in such systems, by means of information theory.

The following two chapters represent the background on the concepts we have used to create our methods to analyse the structure-dynamics relationships in models of complex biological systems. In the second chapter, we present the fundamental ideas from Shannon's information theory and from Kolmogorov complexity. We continue with a generalization of Shannon's theory, named Rényi's information theory, which has recently found significant applications in signal processing, communications engineering and dynamical systems. We describe the results of our theoretical contributions to the field of Rényi's information theory. In chapter three, we present the definition and the properties of the multidimensional scaling visualization method, which we employed in our analysis of the discrete computational model of the human immune system.

In chapter four, we investigate the structure-dynamics relationships in the discrete executable model of the regulation of cytokines in the human immune system. We begin with a brief description of the main elements of the human immune system. We further describe the specific details of the properties and functions of the main agents of the biological system under study. We continue with the computational model that was developed to analyse the dynamics of this biological system. We end the chapter with our contributions and results regarding the analysis and predictions

of the dynamical behaiour of this executable model.

In chapter five, we analyse the random Boolean network model of gene regulatory networks. The most important aspects of random Boolean networks are covered in this chapter. We start with the definition and the properties of the structure of random Boolean networks, emphasizing how we created the classes of structures for our experiments. We describe the salient features of the dynamics of such models. We continue with a unification of these two separate elements, when we present the prior work on the information-theoretic analysis of the structure-dynamics relationships in such models. We end the chapter with our contributions to the study of the structure-dynamics relationships in random Boolean networks. We reserve the last chapter for the discussion of what has been achieved in this thesis.

Chapter 1

Introduction

1.1 Complex systems

Complex systems are present in all areas of scientific enquiry, such as physics, biology, ecology, economics, social science, human societies, engineering [82]. The architecture of complex systems, their function, their properties and the relationships between them constitute the new science of complex systems [8]. In order to understand and describe such systems, we need precise mathematical and computational models that can emulate their structure are replicate their dynamical behaviour. To construct these models, we need experimental methods to investigate naturally occurring complex systems and to collect structural and dynamical information from them. After the model construction phase, computational tools need to be developed that can analyse the data given by such models, produce meaningful predictions and test hypotheses.

As there is no universal definition of complexity for complex systems, we can explain their organizing principles and function through their features [66]. This gives an intuition into why they are termed complex. The salient features of complex systems are: *nonlinearity*, *presence of chaos*, *a vast number of interacting elements*, *feedback*, *self-organization*, *pattern formation*, *robustness* and *adaptability* to the environment, *decentralized control*, *emergence*, a hierarchy of *multiple levels of organization*, *evolvability* [66], [8]. Complex systems can be described as a multitude of connected elements or agents, operating together as a whole, whose behaviour as an entire organism is different from that of the individual agent. Thus, the global behaviour is given by the interactions between the agents and not solely by the be-

haviour of a particular agent. This property is known as *emergence* or *emergent behaviour*. The concepts of *pattern formation*, *self-organization* and *decentralized control* are closely related to emergence. Complex systems are distributed, self-organized systems, without having a central controller to steer the global behaviour of the collection of agents. This is achieved by the self-organization of the agents, locally, which produces global patterns of dynamical behaviour of the overall system. *Multiple levels of organization* refers to the different levels of detail in the description of the system. For example, biological systems are inherently multiscale systems, organized on different levels, such as genes, proteins, molecules, cells, tissues, organs, the human body. At each level, the system can be represented as one type of complex system. Thus, it is very important to model the system at an appropriate level of detail, according to the research question that is being addressed [46].

Complex systems research needs to integrate knowledge and methods from all fields of science and engineering, to understand the complexity of such systems and to create a cohesive body of knowledge for their analysis and synthesis. This endeavour motivates the development of a new science, that of complex systems, with a solid theoretical, computational and experimental foundation. It is no longer enough to study simple systems, in their separate scientific domains, but an integrative approach is necessary, which takes knowledge from disparate fields of research, to understand the naturally occurring complex systems. The aims are to understand and characterize such systems, to create better performing and more efficient engineered systems and to uncover the general principles that govern the structure and the function of complex systems and the relationships between the two [9].

In the pursuit of establishing the science of complex systems, research involves several challenges. They include the description of complex systems, model creation and computer simulations, measuring the complexity of such systems and the discovery of the universal laws that govern the structure and function of such systems, as, for example, the laws of classical mechanics. Research in complex systems tries to understand how these systems are structured, how to describe their structural characteristics and their dynamical behaviour, how to represent the relationship between their structure and their function and how they have evolved to have a certain architecture and dynamical behaviour. The focus is both on investigating specific complex systems, as well as discovering the general laws that govern their function [9].

1.2 Complex networks

Complex systems from multiple scientific areas share the same global characteristics and general laws of behaviour. They differ in the structural and functional details of the individual agents that interact together to create the system [9]. In many cases, eliminating the specific details of the agents from the analysis and focusing on the complex interactions between these elements suffice to obtain a systems level understanding of the dynamics and the structure-dynamics relationships in such systems [107], [79]. As a result, understanding complex systems essentially means understanding the influence between the topology of a complex system, that is, a complex network, and the dynamics that arise on that topology [10].

Some examples of complex networks from the social sciences include social networks of friendships and business relationships, company directors and academic coauthorship of papers. Other related complex networks examples are the information networks of citations of research articles, the World Wide Web, peer-to-peer networks and relations between words in a thesaurus. From a technological point of view, complexity can be found in networks of distribution, such as the electrical power grid, airline routes, roads, railway systems, telephone networks, postal service, the Internet as the physical network of connectivity between computers and other communication devices and in independent groups of networked computers. Biology is an especially rich area of complex networks, such as gene regulatory networks, metabolic pathways, protein interaction networks, vascular networks, neuronal networks, ecosystems and food webs [79].

A complex network represents a graph with a large number of nodes or vertices, which interact in intricate patterns given by the wiring architecture of the network and the dynamical behaviour of each individual node. The study of complex networks presents several challenges, such as the intricate connectivity patterns, the different types of the nodes and those of the connections between them, the models of the nodes, which may be nonlinear dynamical systems, and the complexity of the entire system. The wiring of the nodes may not always be static, but, it may change and evolve in time. The nonlinearity of the dynamical behaviour of the agents signifies that their behaviour is complicated to describe, which increases the difficulty of analysing the system globally. The system's complexity may increase due to the interactions of the nodes of the network, through the connections between them. The ways they interact to increase the complexity of the system are unknown, as

well as how much the complexity of each node or connection, increases the complexity of the entire system [107].

The ultimate goals of analysing complex networks is to uncover the relationship between their structure and their function and to predict the dynamical behaviour of networks with a given type of structure [79]. As a first approach to make such analyses tractable, the dynamics of the complex network are simplified and their structure and their dynamics are investigated separately [79], [16]. As can be seen from the examples mentioned above, real-world networks can be investigated by simplifying the dynamics of their constituent elements and focusing on a more complicated structure. The connectivity patterns are more elaborate and the dynamical behaviour that arises on this more complicated structure is investigated [107].

Graph theory represents the mathematical and computational paradigm used to describe and analyse *the structure of complex networks*. The network is modelled as a collection of nodes, which can be linked in various patterns. Some of the more simple types of connections are the mathematical model of a fully-connected graph, a regular lattice and a random graph, also known as the Erdős-Rényi model [34], [35]. Empirical studies of real-world networks have shown that the scale-free network model [11] and the small-world network model [113] better capture the structural properties of real-world networks than the above mentioned ones do. The first class of models are important for theoretical study, for the development of new analysis methods and as a null hypothesis against which to compare the features of real-world networks. The second class of models are important because they have been observed empirically in several types of real-world complex networks [79]. Empirical graph-theoretic measures of the structural properties of real-world networks differ significantly from those computed for the random graph model [1]. This indicates that the complexity of real-world networks is greater than what a random model can capture and that more sophisticated network models need to be developed to understand their properties.

The most common graph-theoretic measures that characterize the structural properties of complex networks are *the node degree*, *the degree distribution*, *the clustering coefficient*, *the motifs*, *the modularity index*, *the community structure*, *the shortest path length*, *the diameter*, *the node-betweenness* and *the edge-betweenness* [16]. *The node degree* refers to the number of edges that connect to a given node. If the network is directed, there is a distinction between the node in-degree and the node out-degree. *The node in-degree* refers to the incoming edges towards that node

and *the node out-degree* refers to the outgoing edges from that node. They may or may not be equal. The only condition imposed on the node degrees of a network is that the overall sum of the in-degrees is equal to the sum of the out-degrees of the network. The degree distribution refers to the probability distribution of the node degrees, that is, each node of the network has its degree drawn from a given probability distribution. If the network is directed, an in-degree and an out-degree distribution exist for the network. These distributions can be identical or they can be different, even of different types, such as a Poisson distribution and a scale-free distribution. One condition is required, that the overall sum of the in-degrees be equal to the sum of the out-degrees of the network.

The clustering coefficient [113], [38] measures the ratio of the number of the actual connections between the neighbours of a node, to the total possible connections between them. *The motifs* represent small connectivity patterns, which perform a specific function and whose number is greater than what would appear by chance in the network [4]. *The modularity index* represents a measure of the community structure of a network [44], [83]. Higher values of the modularity index indicate that there are several regions of the network, where nodes have a high density of connections between them, while these regions are loosely connected between them. *The shortest path length* between two nodes represents the number of edges that connect the two nodes through other intermediate nodes, such that all nodes are visited only once. *The diameter* of a graph represents the maximum of the shortest path lengths between all pairs of nodes of a network [16]. For each node of the network, *the node-betweenness* measures the sum of the fractions of the shortest paths between any pair of nodes, which contain that node [41], [16]. For each edge of the network, *the edge-betweenness* is equal to the number of shortest paths that contain that edge [83], [16]. The betweenness measures indicate the centrality of nodes and edges and are extremely useful in detecting community structure in networks.

The dynamical behaviour of complex networks is investigated by taking a particular type of structure and implementing dynamical processes on that structure. The networks can have any of the topologies mentioned above. Some of the dynamical processes that have been investigated on these networks include phase transitions of networks, information propagation in networks, for example, disease and rumor spreading in social networks and computer virus spreading in computer networks, searching algorithms on networks, discrete dynamical processes on networks, such as random Boolean networks and cellular automata, and applications of percolation

theory to the analysis of the dynamical robustness of networks [79].

The relationship between the structure and the dynamics of complex networks, its general laws and its properties have not been studied extensively. Little is known about how changes in the structure produce changes in the dynamics, which classes of structures relate to which classes of dynamics and how to define these types of classes. The question of what amount of change in the structure of a network is needed to achieve a desired outcome for the dynamical behaviour pertains to the synthesis of new complex networks. Other questions are whether or not different structures produce different dynamical behaviour and how to quantify the difference of two structures from the same class and from different classes of structures and how this difference translates into the difference of the dynamics. In addition, the information processing capabilities of a complex network are closely related to the relationship between its structure and its dynamics. Information transmission in a network is shaped by its structure and by the dynamical features of the agents of the network. Classification of complex networks can be performed in terms of how information is processed and transmitted within the network. And, finally, how these theoretical results developed on models of complex networks are relevant for studying real-world networks and what knowledge about these systems the theoretical framework of structure-dynamics relationships can yield.

Information theory has recently been proven successful in investigating the relationships between the structure and the dynamics of complex networks. Initially, it was developed as a mathematical framework for the reliable transmission of information over channels with errors. However, its applications are much broader, outside of the field of communications engineering, to the study of complex systems and networks. The novel view on complex systems and networks is that they are information processing systems. Thus, information theory is the natural framework to investigate the structure, the function and the information processing features of such systems and to provide answers to the research questions mentioned previously. Information theory offers not only the possibility of finding structural information from the dynamics, but also the methods to quantify how information propagates in the network. It brings insight into how the network functions, by uncovering how information propagates between the nodes of the network, how patterns of structure affect patterns of dynamics. Information transmission between the elements of a network links the dynamics to the structure of the system, because the patterns of connectivity restrict the patterns of information propagation, which are measured

from the dynamics. Information cannot propagate everywhere in the network, but it is shaped by the topology of the network and by the dynamical features of its agents.

1.3 Systems biology

Biological systems are prime examples of complex systems and complex networks. In recent years, the new paradigm of *systems biology* has grown into a very important scientific discipline to tackle the complexity of such systems. The aim of systems biology is to investigate biological systems at a global level, by integrating diverse types of information from different levels of organization, to understand how systems function as a whole [40]. Such an undertaking is facilitated by improved experimental techniques, which produce an ever increasing amount of dynamical information about the biological system. Such vast amount of data requires adequate models for its integration, for drawing conclusions and for producing predictions about the dynamics of the system. This is the role of computation in building models of biological systems. Moreover, constructing the model requires data from biological experiments. These are the two parts of an iterative process that improves both, through their mutual influence. Existing knowledge about the behaviour of the system, obtained through real-world measurements, is incorporated into the model, by parameter inference. Based on the updated parameters, the model generates new predictions and hypotheses. In turn, they shape the design of new biological experiments, which provide further insight into the system's behaviour. They guide biologists to pursue the appropriate experiments and to make better choices in the quantities they want to measure experimentally.

Two paradigms of modeling biological systems are mathematical models and computational or executable models [40]. The quantitative prediction of the parameters under investigation pertains to mathematical modeling. The prediction of the states of the system and the events that take place relate to executable modeling. Mathematical models consist of equations that express how parameters evolve in time. Computational or executable models employ a set of instructions that describe what the state of the system consists of and how it changes in time, in response to external events. The former is a more quantitative description of the behaviour of the system, while the latter is a more qualitative, descriptive explanation of its behaviour.

A reactive system represents a collection of concurrent, parallel processes, which mutually affect their states, as the system evolves in time and external events occur.

Parallel processes represent separate entities that change their behaviour simultaneously. Concurrency refers to the case when multiple external events take place simultaneously or when several processes try to modify the same entity at the same time. In addition, it involves the methods of solving the conflicts when they arise in such cases. For large reactive systems, mathematical models become cumbersome to derive and, quite often, are either analytically intractable or computationally difficult to solve to the desired accuracy. Executable models can better implement the features of reactive systems, such as the complex description of the state of the system, the parallel architecture of the constituent processes and their concurrency. As biological systems are a prime example of reactive systems, they are well suited to executable modeling [40].

Chapter 2

Information theory

2.1 Shannon's information theory

In the ground-breaking article [102], Claude Shannon establishes the foundations of communication theory, which refers to the reliable transmission of information over unreliable, error-prone channels. He describes the nature of information in probabilistic terms and gives theorems for its reliable transmission. The entropy and mutual information are the fundamental concepts of Shannon's information theory and are the basic building blocks of all other more sophisticated concepts from this framework [102], [25]. We will present the definitions and properties of these information-theoretic quantities, in the following section.

2.1.1 Entropy and mutual information

A random variable is a mathematical construct used to describe and measure uncertainty. This variable can take values from a specified set, named *the ensemble*, and it has a function associated to it. If the set is discrete, then the variable is *a discrete random variable* and its function is named *a probability mass function.* If the set is continuous, such as the set of real numbers $\mathbb{R}$, then it is *a continuous random variable* and the function associated to it is also continuous. It is named *a probability density function.* The probability mass function and the probability density function are mappings from the ensemble of the random variable to the set of real numbers, with the property that the values these functions can take from this set sum to 1. Each element of the ensemble of a random variable is an event that can occur with a given certainty, measured by the probability associated with the event. These probability

functions assign real numbers to each event, which show how likely the event is to occur. More formally, let X be a random variable, $p_X(x)$ its probability distribution and $\mathcal{E}_X$ its ensemble, such that $p_X : \mathcal{E}_X \to [0\ \ 1]$. Then,

$$\begin{aligned} & p_X(x) \geq 0, \forall x \in \mathcal{E}_X \\ & \sum_{x \in \mathcal{E}_X} p_X(x) = 1. \end{aligned} \tag{2.1}$$

Since we have applied the information-theoretic methods to analyze Boolean networks, which are discrete in nature, we will present here only the discrete version of the fundamentals of Shannon's information theory, [Ch 2 of [25]]. Their continuous counterparts can be found in [Ch 8 of [25]].

Definition 1. ***Entropy***. *Let X be a discrete random variable, with values from a discrete alphabet, $\mathcal{E}_X$, and let $p_X(x)$ be its probability mass function. The entropy of X is defined as*

$$\mathrm{H}(X) = -\sum_{x \in \mathcal{E}_X} p_X(x) \cdot \log p_X(x). \tag{2.2}$$

In our analyses, we used the logarithm to the base 2, which means that the entropy is measured in bits. As we are using the logarithm to the base 2 throughout this thesis, we will omit it and we will just write log, which is to be read $\log_2$.

Property 1. $\mathrm{H}(X) \geq 0$.

Proof.

$$\begin{aligned} \mathrm{H}(X) = -\sum_{x \in \mathcal{E}_X} p_X(x) \cdot \log p_X(x) & \\ 0 \leq p_X(x) \leq 1, \forall x \in \mathcal{E}_X & \\ \Rightarrow \log p_X(x) \leq 0, \forall x \in \mathcal{E}_X & \\ \Rightarrow p_X(x) \cdot \log p_X(x) \leq 0, \forall x \in \mathcal{E}_X & \\ \Rightarrow \sum_{x \in \mathcal{E}_X} p_X(x) \cdot \log p_X(x) \leq 0 & \\ \Rightarrow \mathrm{H}(X) \geq 0. & \end{aligned} \tag{2.3}$$

□

Definition 2. ***Joint entropy***. *Let X and Y be two discrete random variables, with values from two discrete alphabets, $\mathcal{E}_X$ and $\mathcal{E}_Y$, and let $p_X(x)$ and $p_Y(y)$ be their*

individual probability mass functions and $p_{XY}(x, y)$ be their joint probability mass function. The joint entropy of X and Y is defined as

$$\mathrm{H}(X, Y) = -\sum_{x \in \mathcal{E}_X} \sum_{y \in \mathcal{E}_Y} p_{XY}(x, y) \cdot \log p_{XY}(x, y). \tag{2.4}$$

From probability theory [51] we know that

$$\begin{aligned} p_{XY}(x, y) &= p_{X|Y}(x|y) \cdot p_Y(y) = p_{Y|X}(y|x) \cdot p_X(x) \\ p_X(x) &= \sum_{y \in \mathcal{E}_Y} p_{XY}(x, y). \end{aligned} \tag{2.5}$$

Definition 3. ***The fundamental theorem of expectation*** [***Ch 4 [47]***]. *Let X and Y be two random variables, with ensembles $\mathcal{E}_X$ and $\mathcal{E}_Y$, and g a function, such that $Y = g(X)$. Let $\mathbb{E}Y$ denote the expectation of Y. If it exists, then*

$$\mathbb{E}Y = \sum_{x \in \mathcal{E}_X} g(x) \cdot p_X(x), \tag{2.6}$$

where $p_X(x)$ is the probability density function of X, if X is a continuous random variable, or the probability mass function of X, if X is a discrete random variable.

Remark 1. Let X, Y and Z be three discrete random variables, such that Z is equal to the conditional random variable $X|Y = y$, i.e. $Z = (X|Y = y)$. The conditional random variable $X|Y = y$ has ensemble $\mathcal{E}_X$ and is a function of the value y. For each value of y in $\mathcal{E}_Y$, we have a probability mass function defined for all the values x in $\mathcal{E}_X$:

$$p_{X|Y}(x|y) = \mathbb{P}((X|Y = y) = x) = \mathbb{P}(X = x|Y = y), \forall x \in \mathcal{E}_X. \tag{2.7}$$

As a result, for each value of y, we can define an entropy for each such probability mass function:

$$\mathrm{H}(X|Y = y) = -\sum_{x \in \mathcal{E}_X} p_{X|Y}(x|y) \cdot \log p_{X|Y}(x|y). \tag{2.8}$$

Remark 2. For each value of y, we have one such entropy, $\mathrm{H}(X|Y = y)$. If we let $W = \mathrm{H}(X|Y = y) = g(y)$, $\forall y \in \mathcal{E}_Y$, then W is now a function, g, only of y. This is because, in the definition of $\mathrm{H}(X|Y = y)$, we summed out all the values $x \in \mathcal{E}_X$. The function g takes values from $\mathcal{E}_Y$ and transforms the random variable Y, according to the definition 2.8. We are not interested in the characteristics of this function, because it serves just as a notation to make the explanations more straightforward.

As W is a function of the random variable Y, it becomes itself a random variable. Thus, the expectation of W is well defined. According to the fundamental theorem of expectation 2.6, the expectation of W is equal to

$$\mathbb{E}W = \sum_{y \in \mathcal{E}_y} p_Y(y) \cdot g(y). \tag{2.9}$$

Definition 4. ***Conditional entropy***. *The conditional entropy* $\mathrm{H}(X|Y)$ *is defined as the expectation of* W, $\mathbb{E}W$*:*

$$\begin{aligned} \mathrm{H}(X|Y) &= -\sum_{y \in \mathcal{E}_y} p_Y(y) \cdot \sum_{x \in \mathcal{E}_X} p_{X|Y}(x|y) \cdot \log p_{X|Y}(x|y) \\ &= -\sum_{y \in \mathcal{E}_y} \sum_{x \in \mathcal{E}_X} p_{XY}(x,y) \cdot \log p_{X|Y}(x|y). \end{aligned} \tag{2.10}$$

The relative entropy or Kullback-Leibler divergence [65] represents one type of measure of the difference between two probability mass functions defined on the same ensemble of a random variable.

Definition 5. ***Kullback-Leibler divergence***. *Let* X *be a discrete random variable and let* $p_X(x)$ *and* $q_X(x)$ *be two probability mass functions defined on the ensemble of* X, $\mathcal{E}_X$*. The Kullback-Leibler divergence between the two probability mass functions is defined as [65]*

$$\mathrm{D}_{\mathrm{KL}}(p||q) = \sum_{x \in \mathcal{E}_X} p_X(x) \cdot \log \frac{p_X(x)}{q_X(x)}. \tag{2.11}$$

If the two probability mass functions are identical, then $\mathrm{D}_{\mathrm{KL}}(p||q) = 0$, otherwise $\mathrm{D}_{\mathrm{KL}}(p||q) > 0$.

Definition 6. ***Conditional Kullback-Leibler divergence***. *Let* X, Y *be two discrete random variables and let* $p_{XY}(x,y)$ *be their joint probability mass function defined on their joint ensemble:* $(x,y) \in \mathcal{E}_X \times \mathcal{E}_Y$*. Let* $p_{X|Y}(x|y)$ *and* $q_{X|Y}(x|y)$ *be the conditional probability mass functions defined on the ensemble of* X, $\mathcal{E}_X$*. Then, the conditional Kullback-Leibler divergence between the two probability mass functions,* p *and* q*, is defined as [25]*

$$\mathrm{D}^*_{\mathrm{KL}}((p(X|Y) \parallel q(X|Y)) = \sum_{x \in \mathcal{E}_X} \sum_{y \in \mathcal{E}_Y} p_{XY}(x,y) \cdot \log \frac{p_{X|Y}(x|y)}{q_{X|Y}(x|y)}. \tag{2.12}$$

Remark 3. By the same arguments as in Remark 1, we will prove in the following paragraph how this equation can be derived from the Kullback-Leibler divergence. Let X, Y and Z be three discrete random variables, such that Z is equal to the conditional random variable $X|Y = y$, i.e. $Z = (X|Y = y)$. The conditional random variable $X|Y = y$ has ensemble $\mathcal{E}_X$ and is a function of the value y. For each value of y in $\mathcal{E}_Y$, we can have two conditional probability mass functions, $p_{X|Y}(x|y)$ and $q_{X|Y}(x|y)$, defined on the ensemble of X, $\mathcal{E}_X$. Let $\mathbb{P}_p(Z = x)$ be the probability that the random variable Z takes the value x, with respect to the probability mass function p. Let $\mathbb{P}_q(Z = x)$ be the probability that the random variable Z takes the value x, with respect to the probability mass function q. We have that $\mathbb{P}_p(Z = x) \neq \mathbb{P}_q(Z = x)$, because p and q are two probability mass functions that assign different probabilities to the same events. The role of any type of divergence is to measure the discrepancy between two probability distributions defined on the same ensemble. Then,

$$\begin{aligned} p_{X|Y}(x|y) &= \mathbb{P}_p((X|Y = y) = x) = \mathbb{P}_p(X = x|Y = y), \forall x \in \mathcal{E}_X \\ q_{X|Y}(x|y) &= \mathbb{P}_q((X|Y = y) = x) = \mathbb{P}_q(X = x|Y = y), \forall x \in \mathcal{E}_X. \end{aligned} \tag{2.13}$$

As a result, for each value of y, we can define a Kullback-Leibler divergence between the two probability mass functions p and q:

$$\mathrm{D_{KL}}(p_{X|Y}(X|Y = y)||q_{X|Y}(X|Y = y)) = \sum_{x \in \mathcal{E}_X} p_{X|Y}(x|y) \cdot \log \frac{p_{X|Y}(x|y)}{q_{X|Y}(x|y)}.$$

$\mathrm{D_{KL}}(p_{X|Y}(X|Y = y)||q_{X|Y}(X|Y = y))$ is now a function of the random variable Y. Thus, it is also a random variable. We are interested in obtaining a value for the divergence, which indicates how different two probability mass functions are. Therefore, we will take this value to be the expectation with respect to Y of the random variable $\mathrm{D_{KL}}(p_{X|Y}(X|Y = y)||q_{X|Y}(X|Y = y))$. That is, by the fundamental theorem of expectation,

$$\begin{aligned} \mathrm{D^*_{KL}}(p(X|Y) \parallel q(X|Y)) &= \mathbb{E}_Y \, \mathrm{D_{KL}}(p_{X|Y}(X|Y = y)||q_{X|Y}(X|Y = y)) = \\ &= \sum_{y \in \mathcal{E}_Y} p_Y(y) \cdot \sum_{x \in \mathcal{E}_X} p_{X|Y}(x|y) \cdot \log \frac{p_{X|Y}(x|y)}{q_{X|Y}(x|y)} \\ &= \sum_{x \in \mathcal{E}_X} \sum_{y \in \mathcal{E}_Y} p_{XY}(x, y) \cdot \log \frac{p_{X|Y}(x|y)}{q_{X|Y}(x|y)}. \end{aligned} \tag{2.14}$$

The following paragraphs explain the properties of convex functions, which are needed to present the properties of information divergences. Chapter 1 from the book by D.S. Mitrinović, [76], is a reference for the definition and the properties of a convex function and for Jensen's inequality for convex functions.

Definition 7. ***Convex function.*** *A function f, defined on a closed interval $[a,b] \subset \mathbb{R}$, $f : [a,b] \to \mathbb{R}$, is convex if and only if*

$$f(\lambda \cdot x + (1-\lambda) \cdot y) \leq \lambda \cdot f(x) + (1-\lambda) \cdot f(y), \forall x, y \in [a,b], \forall \lambda \in [0,1]. \tag{2.15}$$

Property 2. *A function $f : [a,b] \to \mathbb{R}$ is convex on the closed interval $[a,b]$, if and only if its second derivative is nonnegative on the entire interval, i.e. $f''(x) \geq 0, \forall x \in [a,b]$. The proof of this property can be found in [Ch 1 of [76], pp. 17].*

Theorem 1. ***Jensen's inequality [55].*** *Let $f : [a,b] \to \mathbb{R}$ be a continuous and convex function on the interval $[a,b]$ and let $c_1, c_2, \ldots, c_n$ be arbitrary positive numbers, then the following inequality holds*

$$f\left(\frac{\sum_{i=1}^{n} c_i \cdot x_i}{\sum_{i=1}^{n} c_i}\right) \leq \frac{\sum_{i=1}^{n} c_i \cdot f(x_i)}{\sum_{i=1}^{n} c_i} \tag{2.16}$$

Theorem 2. ***The log-sum inequality.*** *Let x_i, $y_i \in \mathbb{R}$, $\forall i = 1, \ldots, n$, then the following inequality holds*

$$\sum_{i=1}^{n} x_i \cdot \log \frac{x_i}{y_i} \geq \left(\sum_{i=1}^{n} x_i\right) \cdot \log \left(\frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} y_i}\right). \tag{2.17}$$

Proof. Let $g : (0, +\infty) \to \mathbb{R}$, $g(t) = t \cdot \log t$. Next, we will prove that this function is convex.

$$\begin{aligned} g'(t) &= \log t + t \cdot \frac{1}{t \cdot \ln 2} = \log t + \frac{1}{\ln 2} \\ g''(t) &= \frac{1}{t \cdot \ln 2} > 0. \end{aligned} \tag{2.18}$$

By property 2 $\Rightarrow g(t)$ is a convex function $\Rightarrow$ we can apply Jensen's inequality 1.

$$\Rightarrow g\left(\frac{\sum_{i=1}^{n} c_i \cdot t_i}{\sum_{i=1}^{n} c_i}\right) \leq \frac{\sum_{i=1}^{n} c_i \cdot g(t_i)}{\sum_{i=1}^{n} c_i}$$

$$\Rightarrow \frac{\sum_{i=1}^{n} c_i \cdot t_i}{\sum_{i=1}^{n} c_i} \cdot \log\left(\frac{\sum_{i=1}^{n} c_i \cdot t_i}{\sum_{i=1}^{n} c_i}\right) \leq \frac{\sum_{i=1}^{n} c_i \cdot t_i \cdot \log t_i}{\sum_{i=1}^{n} c_i}$$

$$\Rightarrow \sum_{i=1}^{n} c_i \cdot t_i \cdot \log t_i \geq \left(\sum_{i=1}^{n} c_i \cdot t_i\right) \cdot \log\left(\frac{\sum_{i=1}^{n} c_i \cdot t_i}{\sum_{i=1}^{n} c_i}\right).$$

Let $t_i = \frac{x_i}{y_i}$ and $c_i = y_i$, $\forall i = 1 : n$.

$$\Rightarrow \sum_{i=1}^{n} y_i \cdot \frac{x_i}{y_i} \cdot \log\frac{x_i}{y_i} \geq \left(\sum_{i=1}^{n} y_i \cdot \frac{x_i}{y_i}\right) \cdot \log\left(\frac{\sum_{i=1}^{n} y_i \cdot \frac{x_i}{y_i}}{\sum_{i=1}^{n} y_i}\right)$$

$$\Rightarrow \sum_{i=1}^{n} x_i \cdot \log\frac{x_i}{y_i} \geq \left(\sum_{i=1}^{n} x_i\right) \cdot \log\left(\frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} y_i}\right). \tag{2.19}$$

□

Property 3. $\mathrm{D_{KL}}(p||q) \geq 0$.

Proof.

$$\mathrm{D_{KL}}(p||q) = \sum_{x \in \mathcal{E}_X} p_X(x) \cdot \log\frac{p_X(x)}{q_X(x)}.$$

This equation is identical to the left term in the log-sum inequality 2.17.

$$\Rightarrow \sum_{x\in\mathcal{E}_X} p_X(x)\cdot\log\frac{p_X(x)}{q_X(x)} \geq \left(\sum_{x\in\mathcal{E}_X} p_X(x)\right)\cdot\log\left(\frac{\sum\limits_{x\in\mathcal{E}_X} p_X(x)}{\sum\limits_{x\in\mathcal{E}_X} q_X(x)}\right)$$

$$\Rightarrow \sum_{x\in\mathcal{E}_X} p_X(x)\cdot\log\frac{p_X(x)}{q_X(x)} \geq 1\cdot\log 1 = 0$$

$$\Rightarrow \mathrm{D_{KL}}(p||q) \geq 0, \tag{2.20}$$

where $\sum_{x\in\mathcal{E}_X} p_X(x) = 1$ and $\sum_{x\in\mathcal{E}_X} q_X(x) = 1$, since $p_X(x)$ and $q_X(x)$ are probability mass functions. □

The mutual information between two random variables represents the amount of information that is shared between the two variables. It is defined as the Kullback-Leibler divergence between their joint probability mass function and the product of their marginal probability mass functions.

Remark 4. The marginal probability mass function of a random variable represents the individual probability mass function of that variable. Given the joint distribution of two random variables, the marginal distribution of one variable is obtained from the joint distribution, by summing out the other variable:

$$p_X(x) = \sum_{y\in\mathcal{E}_Y} p_{XY}(x,y). \tag{2.21}$$

Definition 8. ***Mutual information.*** *Let X and Y be two discrete random variables, with values from two discrete alphabets, $\mathcal{E}_X$ and $\mathcal{E}_Y$, and let $p_X(x)$ and $p_Y(y)$ be their individual probability mass functions and $p_{XY}(x,y)$ be their joint probability mass function. The mutual information between X and Y is defined as*

$$\mathrm{MI}(X,Y) = \sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y} p_{XY}(x,y)\cdot\log\frac{p_{XY}(x,y)}{p_X(x)\cdot p_Y(y)}. \tag{2.22}$$

Property 4. *If two random variables X and Y are independent, then their joint probability mass function is equal to the product of their marginals: $p_{XY}(x,y) = p_X(x)\cdot p_Y(y)$. As a result, the mutual information between two independent random variables is 0.*

Property 5.

$$\begin{aligned}\mathrm{MI}(X,Y) &= \mathrm{H}(X) - \mathrm{H}(X|Y) \\ &= \mathrm{H}(Y) - \mathrm{H}(Y|X) \\ &= \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y). \end{aligned} \tag{2.23}$$

Proof.

$$\begin{aligned}\mathrm{MI}(X,Y) &= \sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y} p_{XY}(x,y)\cdot\log\frac{p_{XY}(x,y)}{p_X(x)\cdot p_Y(y)} \\ \mathrm{H}(X|Y) &= -\sum_{y\in\mathcal{E}_y}\sum_{x\in\mathcal{E}_X} p_{XY}(x,y)\cdot\log p_{X|Y}(x|y) \\ \mathrm{H}(Y|X) &= -\sum_{x\in\mathcal{E}_x}\sum_{y\in\mathcal{E}_Y} p_{XY}(x,y)\cdot\log p_{Y|X}(y|x) \\ \mathrm{H}(X,Y) &= -\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y} p_{XY}(x,y)\cdot\log p_{XY}(x,y) \\ \mathrm{H}(X) &= -\sum_{x\in\mathcal{E}_X} p_X(x)\cdot\log p_X(x) \\ \mathrm{H}(Y) &= -\sum_{y\in\mathcal{E}_Y} p_Y(y)\cdot\log p_Y(y). \end{aligned}$$

From probability theory [51] we know that

$$\begin{aligned} p_{XY}(x,y) &= p_{X|Y}(x|y)\cdot p_Y(y) = p_{Y|X}(y|x)\cdot p_X(x) \\ p_X(x) &= \sum_{y\in\mathcal{E}_Y} p_{XY}(x,y). \end{aligned} \tag{2.24}$$

$$\begin{aligned}\mathrm{H}(X) - \mathrm{H}(X|Y) &= -\sum_{x\in\mathcal{E}_X} p_X(x)\cdot\log p_X(x) + \sum_{y\in\mathcal{E}_y}\sum_{x\in\mathcal{E}_X} p_{XY}(x,y)\cdot\log p_{X|Y}(x|y) \\ &= \sum_{x\in\mathcal{E}_X}\left[-p_X(x)\cdot\log p_X(x) + \sum_{y\in\mathcal{E}_y} p_{XY}(x,y)\cdot\log p_{X|Y}(x|y)\right] \\ &= \sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}\left[-p_{XY}(x,y)\cdot\log p_X(x) + p_{XY}(x,y)\cdot\log\frac{p_{XY}(x,y)}{p_Y(y)}\right] \\ &= \sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y} p_{XY}(x,y)\cdot\log\frac{p_{XY}(x,y)}{p_X(x)\cdot p_Y(y)} = \mathrm{MI}(X,Y). \end{aligned} \tag{2.25}$$

$$\Rightarrow \mathrm{MI}(X,Y) = \mathrm{H}(X) - \mathrm{H}(X|Y). \tag{2.26}$$

Similarly, we can prove that $\mathrm{MI}(X,Y) = \mathrm{H}(Y) - \mathrm{H}(Y|X)$. We will now prove that $\mathrm{MI}(X,Y) = \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y)$.

$$\begin{aligned}
\mathrm{H}(X,Y) &= -\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y} p_{XY}(x,y)\cdot\log p_{XY}(x,y) \\
&= -\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y} p_{XY}(x,y)\cdot\log\left[p_{X|Y}(x|y)\cdot p_Y(y)\right] \\
&= -\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y} p_{XY}(x,y)\cdot\log p_{X|Y}(x|y) - \sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y} p_{XY}(x,y)\cdot\log p_Y(y) \\
&= \mathrm{H}(X|Y) - \sum_{y\in\mathcal{E}_Y} p_Y(y)\cdot\log p_Y(y) \\
&= \mathrm{H}(X|Y) + \mathrm{H}(Y).
\end{aligned} \tag{2.27}$$

$$\begin{aligned}
&\Rightarrow -\mathrm{H}(X|Y) = \mathrm{H}(Y) - \mathrm{H}(X,Y) \\
&\Rightarrow \mathrm{H}(X) - \mathrm{H}(X|Y) = \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y) \\
&\Rightarrow \mathrm{MI}(X,Y) = \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y).
\end{aligned} \tag{2.28}$$

□

Property 3 shows that the Kullback-Leibler divergence between two probability mass functions is always greater or equal to 0. As it is a Kullback-Leibler divergence, the mutual information between two random variables will also be greater or equal to 0. The lower bound of the mutual information is 0, but the upper bound can be any constant up to ∞. Therefore, the mutual information can have a wide range of variation and it can belong to different intervals of values for different systems, but with the same meaning. As such, when conducting a comparative analysis of the behaviour of two dynamical systems or in a classification task, employing the mutual information to distinguish between two objects can be misleading. In such cases, normalized values of the mutual information provide a valid method of comparison between two distinct systems. In our case, the normalized version that gave the best results is [74]:

Definition 9. ***Normalized mutual information****. The normalized mutual information between two random variables X and Y is defined as*

$$\mathrm{nMI}(X,Y) = \frac{\mathrm{MI}(X,Y)}{\mathrm{H}(X,Y)} = \frac{\mathrm{H}(X)+\mathrm{H}(Y)-\mathrm{H}(X,Y))}{\mathrm{H}(X,Y)} = \frac{\mathrm{H}(X)+\mathrm{H}(Y)}{\mathrm{H}(X,Y)} - 1. \tag{2.29}$$

The normalized mutual information has the following properties:

$$\begin{aligned}
&\text{If } X = Y \Rightarrow \mathrm{nMI}(X,Y) = \frac{2 \cdot \mathrm{H}(X)}{\mathrm{H}(X,X)} - 1 = 1. \\
&\text{If } X, Y \text{ are independent, i.e. } p(x,y) = p(x) \cdot p(y) \\
&\quad \Rightarrow \mathrm{H}(X,Y) = \mathrm{H}(X) + \mathrm{H}(Y) \\
&\quad \Rightarrow \mathrm{nMI}(X,Y) = \frac{\mathrm{H}(X) + \mathrm{H}(Y)}{\mathrm{H}(X,Y)} - 1 = 0.
\end{aligned} \tag{2.30}$$

As a conclusion, the normalized mutual information values belong to the interval $[0\ \ 1]$.

2.2 Generalizations of Shannon's information theory

Motivation

Characterizing the structure-dynamics laws in complex systems and networks brings further understanding of how they function as a whole, at the global level. It improves on the understanding of how structural changes affect the dynamical behaviour. The application of such knowledge is in achieving certain desired dynamics in the synthesis process of such systems. If we study the structure and the dynamics separately, without investigating their structure-dynamics relationship, it would be impossible to design new complex systems, that is, creating a type of structure, to obtain a desired dynamical behaviour.

Structural information is embedded in the dynamics of complex systems and networks. Our aim is to reveal this information, from the dynamics, by means of information theory. To this end, we improve on the capabilities of the mutual information of detecting information transfer in such systems. We introduce other equations that are more advanced than the mutual information, to quantify the amount of information flow in these systems. These measures employ random vectors that represent the environment and they also combine the system's dynamical behaviour in intelligent ways, such as to uncover as much hidden structural information as possible. These more complicated equations offer more possibilities to study the structure-dynamics relationships in complex systems and networks, than what mutual information offers. They are still under development and novel applications are being found for them.

Shannon's information theory contains extensions of the mutual information for analysing how information propagates within a complex network and a complex system. Moreover, Shannon's information theory has been generalized to Rényi's information theory, named after Alfred Rényi, who first extended Shannon's entropy to a more complete version, which includes the classical entropy as a limiting case. The role of the classical mutual information has been described in various sections throughout this thesis. These information-theoretic extensions improve on the ability of the mutual information to discover the patterns of information transmission. They have essentially the same usage as the mutual information, in the effort to uncover the structure-dynamics relationships in complex networks. However, these extensions provide more sophisticated means of answering the same questions as the mutual information does. These new measures of information offer rich opportunities of finding structural information hidden in the dynamics.

The most general form of the information theoretic equations described here is in the case when these quantities are computed for random vectors. We will use the random vector representation for our study. But, for clarity and as proof of concept, we present these information-theoretic generalizations in the case of random variables. They are identical if the random variables are replaced with random vectors, because the concepts involved in their definition do not change. For example, the entropy of multiple random variables, that is, of a random vector, is defined by replacing the univariate probability mass function from the definition of the entropy, with a multivariate one [25]. The same applies for the divergence. As all the equations are based on the Rényi α-divergence and on the conditional Rényi α-divergence, the principle described above applies to all the information-theoretic equations presented in this section.

The analysis of the dynamics of complex systems requires random vectors, because the past states of the variables of interest need to be taken into consideration, as well as the influence of other elements, which represent the environment. As we have seen in the introductory section to the definitions and properties of complex systems, the interactions between the elements of the system are extremely complex and are very difficult to characterize. Combining the information about the past states of elements of the system, with the influence of the environment enhances the more traditional analysis of using the mutual information and even the time-delayed mutual information. Because of the correlations in the system, often past states influence the present state of an element. Thus, it may contain valuable information regarding the present activity of that element, as well as regarding its patterns of connectivity within the network. The structural properties are hidden in the dynamics through the way they restrict the possible dynamics of the elements of the systems, that is, through the way they introduce correlations in the dynamical states of the elements of the complex system. The presence and the patterns of these correlations can be found through such statistical analyses that merge the types of features described above.

Our contributions in this field are the derivation of a new equation from Rényi's information theory, which we term the partial Rényi transfer entropy [98] and the alternative derivation of the partial Rényi mutual information [98], which was first introduced in [88]. Unlike the authors of [88], who state the definition without deriving it, we provide logical arguments to our choice of derivation and to the validity of this definition. We present the prior work on this topic, to create the context in which

these new equations were derived. We discuss existing means from Shannon's and Rényi's information theories, which quantify the transfer of information in complex networks and systems. To prove the practical application of our methods, we have successfully characterized structural parameters from the dynamical behaviour of an autoregression systems of order 1. As another example, the random Boolean network model could be a first step in understanding how these generalizations perform on models of complex networks and systems.

Rényi's entropy is a generalization of Shannon's entropy

In his paper [92], Alfred Rényi generalizes Shannon's entropy and establishes another branch of information theory, named Rényi's information theory. Although this generalization was introduced in 1961, applications of Rényi's information theory have only been found recently and now increased attention is devoted to this subfield of information theory. More applications of Rényi's entropy and divergence have been found in signal processing and communications engineering. The cutoff rate in block coding is a parameter involved in the upper bound of the average probability of error of a code [26]. The author of [26] introduces a parameter β in these upper bounds and links these generalized β-cuttoff rates of a discrete memoryless channel with Rényi's entropy and divergence. Rényi's entropy is applied to the single-input single-output blind deconvolution problem for linear channels [36]. The authors of [90] find analytical expressions for the Rényi divergence rate and the Rényi entropy rate, in the case of finite-alphabet Markov sources, which are time-invariant and of arbitrary order. The authors of [12] employ Rényi's entropy to measure the amount of information and the complexity of signals. They consider the time-frequency representation of a signal, that is, the two-dimensional function which has two variables, the time and the frequency, as a two-dimensional probability distribution. They use this function as the distribution in the definition of Rényi's entropy. Other applications of Rényi's entropy include machine learning [77], [89], adaptive system training [37] and clustering [45].

In complex systems science, research has been focused more on developing the theoretical foundations of Rényi's information theory. The authors of [56] introduce the Rényi transfer entropy and apply it to the US, Europe and Asia-Pacific financial markets, to study the direction and amount of information flow between them. The conditional Rényi mutual information is introduced by [88] and we define the partial

Rényi transfer entropy in [98]. However, these generalized information measures have not yet been applied to large complex systems and networks. This remains a very important direction for future research, as these measures show great promise in detecting the structure-dynamics relationships in complex systems and networks.

In order to generalize Shannon's entropy, Rényi uses the idea of incomplete random variables and incomplete probability distributions. To explain the generalization of Shannon's entropy, we use the mathematical notation of [92]. Let ξ be an incomplete discrete random variable and $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ be the collection of probabilities assigned to n events that are modelled by the random variable ξ. Then, this probability distribution is termed incomplete, if

$$0 < \sum_{k=1}^{n} p_k \leq 1. \tag{2.31}$$

Let $W(\mathcal{P}) = \sum_{k=1}^{n} p_k$ be the weight of the distribution $\mathcal{P}$. If $W(\mathcal{P}) = 1$, then the random variable and distribution are termed complete or ordinary. The complete or ordinary information-theoretic equations can be obtained from their incomplete versions, by replacing the weight of the distribution with the value 1. Then, Shannon's entropy for incomplete random variables becomes:

$$\mathrm{H}_1[\mathcal{P}] = \frac{\displaystyle\sum_{k=1}^{n} p_k \cdot \log_2 \frac{1}{p_k}}{\displaystyle\sum_{k=1}^{n} p_k}. \tag{2.32}$$

For generalized or incomplete distributions, Rényi's entropy of order α becomes:

$$\mathrm{H}_\alpha[\mathcal{P}] = \frac{1}{1-\alpha} \cdot \log_2 \left(\frac{\displaystyle\sum_{k=1}^{n} p_k^\alpha}{\displaystyle\sum_{k=1}^{n} p_k} \right). \tag{2.33}$$

Rényi extends the axioms satisfied by Shannon's entropy to such distributions, in the form of five postulates. In the theorem 1 of [92], he proves that the only form of the entropy that satisfies these newly defined postulates is Shannon's entropy for incomplete probability distributions. The postulate 5 represents the mean-value property of the entropy and it uses the arithmetic mean in its definition. In the postulate 5, the author states that, if $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ and $\mathcal{Q} = \{q_1, q_2, \ldots, q_n\}$

are two generalized distributions, such that $W(\mathcal{P}) + W(\mathcal{Q}) \leq 1$, and the union of $\mathcal{P}$ and $\mathcal{Q}$ is defined as $\mathcal{P} \cup \mathcal{Q} = \{p_1, p_2, \ldots, p_m, q_1, q_2, \ldots, q_n\}$, then

$$\mathrm{H}[\mathcal{P} \cup \mathcal{Q}] = \frac{W(\mathcal{P}) \cdot \mathrm{H}[\mathcal{P}] + W(\mathcal{Q}) \cdot \mathrm{H}[\mathcal{Q}]}{W(\mathcal{P}) + W(\mathcal{Q})}. \tag{2.34}$$

Then, he changes the arithmetic mean to a generalized mean, which involves a strictly monotonic and continuous function and gives the postulate $5'$. Here, he states that, if there exist $\mathcal{P}$ and $\mathcal{Q}$, such that $W(\mathcal{P}) + W(\mathcal{Q}) \leq 1$, then there exists a strictly monotonic and continuous function $y = g(x)$, which has the inverse function denoted as $x = g^{-1}(y)$, such that

$$\mathrm{H}[\mathcal{P} \cup \mathcal{Q}] = g^{-1}\left[\frac{W(\mathcal{P}) \cdot g(\mathrm{H}[\mathcal{P}]) + W(\mathcal{Q}) \cdot g(\mathrm{H}[\mathcal{Q}])}{W(\mathcal{P}) + W(\mathcal{Q})}\right]. \tag{2.35}$$

He proves that only two functions in this generalized mean are possible, such that the five postulates are satisfied: $g(x) = a \cdot x + b$, with $a \neq 0$, leads to Shannon's entropy and $g(x) = 2^{(\alpha-1)\cdot x}$, with $\alpha > 0$ and $\alpha \neq 1$, leads to Rényi's entropy. That is, in the first case, the postulate $5'$ is equal to the postulate 5. The only entropy that satisfies the postulates $1, 2, 3, 4$ and 5 is Shannon's entropy, as proven by the theorem 1. In the second case, the only entropy that satisfies the postulates $1, 2, 3, 4$ and $5'$ is Rényi's entropy, as proven by the theorem 2. Moreover,

$$\lim_{\alpha \to 1} \mathrm{H}_\alpha[\mathcal{P}] = \mathrm{H}_1[\mathcal{P}]. \tag{2.36}$$

The above explanations are the reasons why Rényi's entropy is a generalizations of Shannon's entropy.

Prior work on probabilistic information-theoretic equations and their generalizations

The first generalization of the mutual information is *the transfer entropy*, introduced in [101]. As with the mutual information, it involves two stochastic processes, X and Y, but, in addition, it has a conditioning variable, which makes this measure a directed one. The transfer entropy from the process X to the process Y measures the shared information between the present state of X and the past states of Y, conditioned on the past states of X. It describes how much the history of the process Y affects the process X, at the present moment, given that the history of X is known. The asymmetry of the transfer entropy makes it extremely suitable

for detecting directed coupling in complex systems and networks, where information may flow in only one direction between two connected elements. *The partial transfer entropy* [109] is an improvement on the transfer entropy and a further generalization of the traditional mutual information. Similarly to the transfer entropy, it quantifies the directed exchange of information from the process X to the process Y. But, it is a more powerful measure of information, because it adds the environment, modeled as a stochastic process Z, as another conditioning variable, together with the history of X. It eliminates any possible indirect influences between the processes X and Y, which may take place through indirect connections between them, through the environment. *The partial mutual information* [42] represents another extension of the mutual information. The authors of [42] define this measure as the conditional mutual information between two processes X and Y, given the process Z, which represents the environment. It quantifies the shared information between X and Y that is not contained in Z.

The transfer entropy and the partial mutual information have been generalized to Rényi's information theory, as *Rényi transfer entropy* by [56] and as *the conditional Rényi mutual information* by [88], respectively. The authors of [56] define the Rényi transfer entropy, by replacing the conditional Shannon entropy in the definition of the transfer entropy, with the conditional Rényi entropy. In our work [98], we give a more straightforward derivation of the Rényi transfer entropy. We employ the conditional Rényi α-divergence, which we derive from the Rényi α-divergence. A different version of this conditional divergence is used in [88], to define the conditional Rényi mutual information. The authors do not motivate the choice of this divergence. We bring contributions to this field of generalized information-theoretic equations by deriving the conditional Rényi divergence from the Rényi α-divergence, using probabilistic and logical arguments. We redefine the most important Rényi information-theoretic equations, on the basis of this conditional divergence, in a unified framework. This is the reason why it is extremely important to have a solid proof of the derivation of the conditional Rényi α-divergence. Based on this divergence, the most notable contribution of our work [98] is *the partial Rényi transfer entropy*, as the generalization of the partial transfer entropy to Rényi's information theory.

For the definition and the properties of the information-theoretic equations that we have investigated, we use the mathematical notation of [98]. This notation is identical to the one used in the section 2.1.

Definition 10. ***Entropy***. *Let X be a discrete random variable, with values from a discrete alphabet, $\mathcal{E}_X$, and let $p_X(x)$ be its probability mass function. The entropy of X is defined as*

$$\mathrm{H}(X) = -\sum_{x\in\mathcal{E}_X} p_X(x)\cdot\log p_X(x). \tag{2.37}$$

Definition 11. ***Rényi α-entropy***. *Let X be a discrete random variable, with values from a discrete alphabet, $\mathcal{E}_X$, and let $p_X(x)$ be its probability mass function. Then, Rényi's entropy of order α is defined as [92]*

$$\mathrm{H}_\alpha(X) = \frac{1}{1-\alpha}\log\sum_{x\in\mathcal{E}_X} p_X^\alpha(x). \tag{2.38}$$

Definition 12. ***Kullback-Leibler divergence***. *Let X be a discrete random variable and let $p_X(x)$ and $q_X(x)$ be two probability mass functions defined on the ensemble of X, $\mathcal{E}_X$. The Kullback-Leibler divergence between the two probability mass functions is defined as [65]*

$$\mathrm{D}_{\mathrm{KL}}(p||q) = \sum_{x\in\mathcal{E}_X} p_X(x)\cdot\log\frac{p_X(x)}{q_X(x)}. \tag{2.39}$$

Definition 13. ***Rényi α-divergence***. *Let X be a discrete random variable, with values from a discrete alphabet, $\mathcal{E}_X$. Let $p_X(x)$ and $q_X(x)$ be two probability mass functions for the random variable X. Then, Rényi's divergence of order α between $p_X(x)$ and $q_X(x)$ is equal to [92]*

$$\mathrm{D}_\alpha(p\parallel q) = \frac{1}{\alpha-1}\log\left(\sum_{x\in\mathcal{E}_X}\frac{p_X^\alpha(x)}{q_X^{\alpha-1}(x)}\right) \tag{2.40}$$

The following definition has been presented in the definition 6, along with the arguments for its derivation in the remark 3. We repeat it here for completness, in order to understand the similarities and differences between Shannon's and Rényi's information theories.

Definition 14. ***Conditional Kullback-Leibler divergence***. *Let X, Y be two discrete random variables and let $p_{XY}(x,y)$ be their joint probability mass function defined on their joint ensemble: $(x,y)\in\mathcal{E}_X\times\mathcal{E}_Y$. Let $p_{X|Y}(x|y)$ and $q_{X|Y}(x|y)$ be the conditional probability mass functions defined on the ensemble of X, $\mathcal{E}_X$. Then, the conditional Kullback-Leibler divergence between the two probability mass functions, p*

and q, is defined as [25]

$$\mathrm{D}^{*}_{\mathrm{KL}}((p(X|Y) \parallel q(X|Y)) = \sum_{x \in \mathcal{E}_X} \sum_{y \in \mathcal{E}_Y} p_{XY}(x,y) \cdot \log \frac{p_{X|Y}(x|y)}{q_{X|Y}(x|y)}. \tag{2.41}$$

Using the same probability theoretic arguments as we provided in the remark 3 for the derivation of the conditional Kullback-Leibler divergence, we derive the conditional Rényi α-divergence. We derive this conditional divergence in a different manner than that of the authors of [88], who also introduced a conditional Rényi α-divergence. It has a different expression than our definition of this divergence. Next, we redefine extisting Rényi information theoretic equations, in terms of this alternative version of the conditional Rényi α-divergence, and we introduce a new equation, termed partial Rényi transfer entropy (PRTE).

Definition 15. ***Conditional Rényi α-divergence.*** *Let X, Y be two discrete random variables and let $p_Y(y)$ be the probability mass function defined on the ensemble of Y, $\mathcal{E}_Y$. Let $p_{X|Y}(x|y)$ and $q_{X|Y}(x|y)$ be the conditional probability mass functions defined on the ensemble of X, $\mathcal{E}_X$. Then, the conditional Rényi α-divergence between the two probability mass functions, p and q, is defined as [98]*

$$\mathrm{D}^{*}_{\alpha}(p(X|Y) \parallel q(X|Y)) = \frac{1}{\alpha - 1} \cdot \sum_{y \in \mathcal{E}_Y} p_Y(y) \cdot \log \left(\sum_{x \in \mathcal{E}_X} \frac{p^{\alpha}_{X|Y}(x|y)}{q^{\alpha-1}_{X|Y}(x|y)} \right). \tag{2.42}$$

Remark 5. Let X, Y and Z be three discrete random variables, such that Z is equal to the conditional random variable $X|Y = y$, i.e. $Z = (X|Y = y)$. The conditional random variable $X|Y = y$ has ensemble $\mathcal{E}_X$ and is a function of the value y. For each value of y in $\mathcal{E}_Y$, we can have two conditional probability mass functions, $p_{X|Y}(x|y)$ and $q_{X|Y}(x|y)$, defined on the ensemble of X, $\mathcal{E}_X$. Let $\mathbb{P}_p(Z = x)$ be the probability that the random variable Z takes the value x, with respect to the probability mass function p. Let $\mathbb{P}_q(Z = x)$ be the probability that the random variable Z takes the value x, with respect to the probability mass function q. We have that $\mathbb{P}_p(Z = x) \neq \mathbb{P}_q(Z = x)$, because p and q are two probability mass functions that assign different probabilities to the same events. The role of any type of divergence is to measure the discrepancy between two probability distributions defined on the same ensemble. Then,

$$\begin{aligned} p_{X|Y}(x|y) &= \mathbb{P}_p((X|Y = y) = x) = \mathbb{P}_p(X = x|Y = y), \forall x \in \mathcal{E}_X \\ q_{X|Y}(x|y) &= \mathbb{P}_q((X|Y = y) = x) = \mathbb{P}_q(X = x|Y = y), \forall x \in \mathcal{E}_X. \end{aligned} \tag{2.43}$$

As a result, for each value of y, we can define a Rényi α-divergence between the two probability mass functions p and q:

$$\mathrm{D}_\alpha(p_{X|Y}(X|Y=y) \parallel q_{X|Y}(X|Y=y)) = \frac{1}{\alpha-1}\log\left(\sum_{x\in\mathcal{E}_X}\frac{p^\alpha_{X|Y}(X=x|Y=y)}{q^{\alpha-1}_{X|Y}(X=x|Y=y)}\right).$$

$\mathrm{D}_\alpha(p_{X|Y}(X|Y=y) \parallel q_{X|Y}(X|Y=y))$ is now a function of the random variable Y. Thus, it is also a random variable. We are interested in obtaining a value for the divergence, which indicates how different two probability mass functions are. Therefore, we will take this value to be the expectation with respect to Y of the random variable $D_\alpha(p_{X|Y}(X|Y=y) \parallel q_{X|Y}(X|Y=y))$. That is, by the fundamental theorem of expectation,

$$\begin{aligned}\mathrm{D}^*_\alpha(p(X|Y) \parallel q(X|Y)) &= \mathbb{E}_Y\,\mathrm{D}_\alpha(p_{X|Y}(X|Y=y)||q_{X|Y}(X|Y=y)) = \\ &= \frac{1}{\alpha-1}\cdot\sum_{y\in\mathcal{E}_Y}p_Y(y)\cdot\log\left(\sum_{x\in\mathcal{E}_X}\frac{p^\alpha_{X|Y}(x|y)}{q^{\alpha-1}_{X|Y}(x|y)}\right). \qquad (2.44)\end{aligned}$$

In the theorem 1 of [98], we prove that this divergence tends to the conditional Kullback-Leibler divergence. As a result, all the Rényi information-theoretic equations will tend to their counterparts from Shannon's theory, as they are all defined based on the conditional Rényi α-divergence. This property is very important, because all the Rényi generalizations must include their classical counterparts as limiting cases.

Definition 16. ***Mutual information*** **(MI)**. *Let X and Y be two discrete random variables, with values from two discrete alphabets, $\mathcal{E}_X$ and $\mathcal{E}_Y$, and let $p_X(x)$ and $p_Y(y)$ be their individual probability mass functions and $p_{XY}(x,y)$ be their joint probability mass function. The mutual information between X and Y is defined as the Kullback-Leibler divergence between the joint probability mass function, $p_{XY}(x,y)$, and the product of the marginals, $p_X(x)\cdot p_Y(y)$:*

$$\begin{aligned}\mathrm{MI}(X,Y) &= \mathrm{D}_{\mathrm{KL}}(p_{XY}(x,y) \parallel (p_X(x)\cdot p_Y(y))) \\ &= \sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}p_{XY}(x,y)\cdot\log\frac{p_{XY}(x,y)}{p_X(x)\cdot p_Y(y)}. \qquad (2.45)\end{aligned}$$

Definition 17. ***Rényi mutual information*** **(RMI)**. *Let X and Y be two discrete random variables, with values from two discrete alphabets, $\mathcal{E}_X$ and $\mathcal{E}_Y$, and let $p_X(x)$*

and $p_Y(y)$ be their individual probability mass functions and $p_{XY}(x,y)$ be their joint probability mass function. The Rényi mutual information between X and Y is defined as the Rényi α-divergence between the joint probability mass function, $p_{XY}(x,y)$, and the product of the marginals, $p_X(x) \cdot p_Y(y)$:

$$\begin{aligned}\mathrm{RMI}(X,Y) &= \mathrm{D}_\alpha(p_{XY}(x,y) \parallel (p_X(x) \cdot p_Y(y))) \\ &= \frac{1}{\alpha-1} \cdot \log\left(\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y} \frac{p_{XY}^\alpha(x,y)}{p_X^{\alpha-1}(x)\cdot p_Y^{\alpha-1}(y)}\right). \end{aligned} \tag{2.46}$$

Definition 18. ***Conditional mutual information (*CMI*).*** *Let X, Y and Z be three discrete random variables and let $p_{XY|Z}(x,y|z)$ be the joint probability mass function of X and Y, conditioned on Z. It is defined on their joint ensemble: $(x,y) \in \mathcal{E}_X \times \mathcal{E}_Y$. Let $p_{X|Z}(x|z)$ and $p_{Y|Z}(y|z)$ be the conditional probability mass functions defined on the ensemble of X, $\mathcal{E}_X$ and on the ensemble of Y, $\mathcal{E}_Y$, respectively, conditioned on Z. Then, the conditional mutual information between X and Y, conditioned on Z, is equal to the conditional Kullback-Leibler divergence between $p_{XY|Z}$ and the product of the conditional marginals, $p_{X|Z} \cdot p_{Y|Z}$:*

$$\begin{aligned}\mathrm{CMI}(X,Y|Z) &= \mathrm{D}_{\mathrm{KL}}^*(p_{XY|Z}(x,y|z) \parallel (p_{X|Z}(x|z)\cdot p_{Y|Z}(y|z))) \\ &= \sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}\sum_{z\in\mathcal{E}_Z} p_{XYZ}(x,y,z)\cdot\log\frac{p_{XY|Z}(x,y|z)}{p_{X|Z}(x|z)\cdot p_{Y|Z}(y|z)}. \end{aligned} \tag{2.47}$$

Definition 19. ***Conditional Rényi mutual information (*CRMI*).*** *Let X, Y and Z be three discrete random variables and let $p_{XY|Z}(x,y|z)$ be the joint probability mass function of X and Y, conditioned on Z. It is defined on their joint ensemble: $(x,y) \in \mathcal{E}_X \times \mathcal{E}_Y$. Let $p_{X|Z}(x|z)$ and $p_{Y|Z}(y|z)$ be the conditional probability mass functions defined on the ensemble of X, $\mathcal{E}_X$ and on the ensemble of Y, $\mathcal{E}_Y$, respectively, conditioned on Z. Then, we define the conditional mutual information between X and Y, conditioned on Z, as the conditional Rényi α-divergence between $p_{XY|Z}$ and the product of the conditional marginals, $p_{X|Z} \cdot p_{Y|Z}$:*

$$\begin{aligned}\mathrm{CRMI}(X,Y|Z) &= \mathrm{D}_\alpha^*(p_{XY|Z}(x,y|z) \parallel (p_{X|Z}(x|z)\cdot p_{Y|Z}(y|z))) \\ &= \frac{1}{\alpha-1}\cdot\sum_{z\in\mathcal{E}_Z} p_Z(z)\cdot\log\left(\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}\frac{p_{XY|Z}^\alpha(x,y|z)}{p_{X|Z}^{\alpha-1}(x|z)\cdot p_{Y|Z}^{\alpha-1}(y|z)}\right) \end{aligned} \tag{2.48}$$

Let X and Y be two random vectors and let k be the time lag for X and l be the time lag for Y. A time lag refers to how many past states are taken into the analysis, from the current moment. The current time point is represented by $n+1$. Then, we have that

$$\begin{aligned} X &= [X_{n+1} \;\; X_n \;\; X_{n-1} \;\; \dots \;\; X_{n-k+1}] \\ Y &= [Y_n \;\; Y_{n-1} \;\; \dots \;\; Y_{n-l+1}]. \end{aligned} \tag{2.49}$$

Here, X_i and Y_j are random variables, $\forall i = \{(n+1), n, \dots, (n-k+1)\}$, $\forall j = \{n, (n-1), \dots, (n-l+1)\}$. For clarity of notation, let

$$\begin{aligned} V &= X_{n+1}, \\ W &= [Y_n \;\; Y_{n-1} \;\; \dots \;\; Y_{n-l+1}], \\ U &= [X_n \;\; X_{n-1} \;\; \dots \;\; X_{n-k+1}]. \end{aligned} \tag{2.50}$$

Definition 20. ***Transfer entropy (*TE*).*** *The* TE *was introduced in [101] to improve on the mutual information between two random vectors, X and Y, by eliminating the effect of the indirect influences of past states of one of the vectors. The* $\mathrm{TE}_{Y\to X}(k,l)$ *measures the mutual information between the current state of X, x_{n+1}, and the past l states of Y, $[Y_n \;\; Y_{n-1} \;\; \dots \;\; Y_{n-l+1}]$, given that the past k states of X, $[X_n \;\; X_{n-1} \;\; \dots \;\; X_{n-k+1}]$ are known:*

$$\begin{aligned} \mathrm{TE}_{Y->X}(k,l) &= \mathrm{CMI}(V,W|U) \\ &= \sum_{v\in\mathcal{E}_V}\sum_{w\in\mathcal{E}_W}\sum_{u\in\mathcal{E}_U} p_{VWU}(v,w,u)\cdot\log\frac{p_{VW|U}(v,w|u)}{p_{V|U}(v|u)\cdot p_{W|U}(w|u)} \end{aligned} \tag{2.51}$$

$$\begin{aligned} \Rightarrow \mathrm{TE}_{Y->X}(k,l) &= \mathrm{CMI}(V,W|U) \\ &= \sum_{x_{n+1}}\sum_{y_n}\cdots\sum_{y_{n-l+1}}\sum_{x_n}\cdots\sum_{x_{n-k+1}} p(x_{n+1},y_n,\dots,y_{n-l+1},x_n,\dots,x_{n-k+1})\cdot \\ &\cdot\log\frac{p(x_{n+1},y_n,\dots,y_{n-l+1}|x_n,\dots,x_{n-k+1})}{p(x_{n+1}|x_n,\dots,x_{n-k+1})\cdot p(y_n,\dots,y_{n-l+1}|x_n,\dots,x_{n-k+1})}. \end{aligned} \tag{2.52}$$

Definition 21. ***Rényi transfer entropy (*RTE*).*** *The* RTE *is computed from one random vector Y to another random vector X, considering different time lags for each vector, l for Y and k for X. It has the same meaning as the* TE*, but it is computed with the generalized version of the information theoretic equation that is present in the definition of the* TE*. The* RTE *is equal to the conditional Rényi*

mutual information between the present state of X, X_{n+1} and the past l states of Y, given that we know the past k states of X [98]. That is, we have

$$\mathrm{RTE}_{Y->X}(k,l) = \mathrm{CRMI}(V,W|U) = \\ = \frac{1}{\alpha-1}\cdot\sum_{u\in\mathcal{E}_U} p_U(u)\cdot\log\left(\sum_{v\in\mathcal{E}_V}\sum_{w\in\mathcal{E}_W}\frac{p^{\alpha}_{VW|U}(v,w|u)}{p^{\alpha-1}_{V|U}(v|u)\cdot p^{\alpha-1}_{W|U}(w|u)}\right) \tag{2.53}$$

$$\Rightarrow \mathrm{RTE}_{Y->X}(k,l) = \mathrm{CRMI}(V,W|U) = \\ = \frac{1}{\alpha-1}\sum_{x_n}\cdots\sum_{x_{n-k+1}} p(x_n,\ldots,x_{n-k+1})\cdot\log\sum_{x_{n+1}}\sum_{y_n}\cdots \\ \cdots\sum_{y_{n-l+1}}\frac{p^{\alpha}(x_{n+1},y_n,\ldots,y_{n-l+1}|x_n,\ldots,x_{n-k+1}))}{p^{\alpha-1}(x_{n+1}|x_n,\ldots,x_{n-k+1})\cdot p^{\alpha-1}(y_n,\ldots,y_{n-l+1}|x_n,\ldots,x_{n-k+1})}. \tag{2.54}$$

Let X, Y and Z be three random vectors and let k be the time lag for X, l be the time lag for Y and m be the time lag for Z. Then, we have that

$$\begin{aligned} X &= [X_{n+1}\ X_n\ X_{n-1}\ \ldots\ X_{n-k+1}] \\ Y &= [Y_n\ Y_{n-1}\ \ldots\ Y_{n-l+1}] \\ Z &= [Z_n\ Z_{n-1}\ \ldots\ Z_{n-m+1}]. \end{aligned} \tag{2.55}$$

Here, X_i, Y_j, Z_r are random variables, $\forall i = \{(n+1), n, \ldots, (n-k+1)\}$, $\forall j = \{n, (n-1), \ldots, (n-l+1)\}$, $\forall r = \{n, (n-1), \ldots, (n-m+1)\}$. For clarity of notation, let

$$\begin{aligned} V &= X_{n+1}, \\ W &= [Y_n\ Y_{n-1}\ \ldots\ Y_{n-l+1}], \\ U &= [X_n\ X_{n-1}\ \ldots\ X_{n-k+1}\ Z_n\ Z_{n-1}\ \ldots\ Z_{n-m+1}]. \end{aligned} \tag{2.56}$$

Definition 22. ***Partial mutual information (*PMI*)****. The* PMI *was introduced in [42]. It is equal to the conditional mutual information between X and Y, given*

that Z is known:

$$\begin{aligned}
&\mathrm{PMI}(X,Y|Z) = \mathrm{CMI}(X,Y|Z)\\
&= \sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}\sum_{z\in\mathcal{E}_Z} p_{XYZ}(x,y,z)\cdot\log\frac{p_{XY|Z}(x,y|z)}{p_{X|Z}(x|z)\cdot p_{Y|Z}(y|z)}\\
&= \sum_{x_n}\cdots\sum_{x_{n-k+1}}\sum_{y_n}\cdots\sum_{y_{n-l+1}}\sum_{z_n}\cdots\sum_{z_{n-m+1}} p(x_n,\ldots,x_{n-k+1},y_n,\ldots,y_{n-l+1},\\
&z_n,\ldots z_{n-m+1})\cdot\log\frac{p(x_n,\ldots,x_{n-k+1},y_n,\ldots,y_{n-k+1}|z_n,\ldots,z_{n-m+1})}{p(x_n,\ldots,x_{n-k+1}|z_n,\ldots,z_{n-m+1})\cdot}\\
&\overline{\cdot p(y_n,\ldots,y_{n-k+1}|z_n,\ldots,z_{n-m+1})}.
\end{aligned} \tag{2.57}$$

Definition 23. ***Partial Rényi mutual information (*PRMI**).** *The* PRMI *represents the conditional Rényi mutual information between X and Y, given that Z is known [98]:*

$$\begin{aligned}
\mathrm{PRMI}(X,Y|Z) = CRMI(X,Y|Z) &= \mathrm{D}^*_\alpha(p_{XY|Z}(x,y|z)\parallel p_{X|Z}(x|z)\cdot p_{Y|Z}(y|z))\\
&= \frac{1}{\alpha-1}\cdot\sum_{z\in\mathcal{E}_Z} p_Z(z)\cdot\log\left(\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}\frac{p^\alpha_{XY|Z}(x,y|z)}{p^{\alpha-1}_{X|Z}(x|z)\cdot p^{\alpha-1}_{Y|Z}(y|z)}\right).
\end{aligned} \tag{2.58}$$

Definition 24. ***Partial transfer entropy (*PTE*)**. *The* PTE *was introduced in [109] to improve on the transfer entropy, by adding the environment as another conditional variable. With the above notations, the* PTE *is defined as*

$$\begin{aligned}
&\mathrm{PTE}_{Y\to X|Z} = CMI(V,W|U)\\
&= \sum_{v\in\mathcal{E}_V}\sum_{w\in\mathcal{E}_W}\sum_{u\in\mathcal{E}_U} p_{VWU}(v,w,u)\cdot\log\frac{p_{VW|U}(v,w|u)}{p_{V|U}(v|u)\cdot p_{W|U}(w|u)}.
\end{aligned} \tag{2.59}$$

$$\begin{aligned}
&\Rightarrow \mathrm{PTE}_{Y->X|Z}(k,l,m) = \mathrm{CMI}(V,W|U)\\
&= \sum_{x_{n+1}}\sum_{y_n}\cdots\sum_{y_{n-l+1}}\sum_{x_n}\cdots\sum_{x_{n-k+1}}\sum_{z_n}\cdots\sum_{z_{n-m+1}} p(x_{n+1},y_n,\ldots,y_{n-l+1},x_n,\ldots,\\
&x_{n-k+1},z_n\ldots z_{n-m+1})\cdot\log\frac{p(x_{n+1},y_n,\ldots,y_{n-l+1}|x_n,\ldots,x_{n-k+1},z_n\ldots z_{n-m+1})}{p(x_{n+1}|x_n,\ldots,x_{n-k+1},z_n\ldots z_{n-m+1})}\\
&\overline{\cdot p(y_n,\ldots,y_{n-l+1}|x_n,\ldots,x_{n-k+1},z_n\ldots z_{n-m+1})}.
\end{aligned} \tag{2.60}$$

Definition 25. ***Partial Rényi transfer entropy (*PRTE*).*** *In addition to the elements of the* RTE*, the probability mass functions are also conditioned on the environment represented as the random vector Z [98]. With the above notations, we define the* PRTE *as*

$$\mathrm{PRTE}_{Y->X}(k,l,m) = \mathrm{CRMI}(V,W|U) = \\ = \frac{1}{\alpha-1}\cdot\sum_{u\in\mathcal{E}_U} p_U(u)\cdot\log\left(\sum_{v\in\mathcal{E}_V}\sum_{w\in\mathcal{E}_W}\frac{p^{\alpha}_{VW|U}(v,w|u)}{p^{\alpha-1}_{V|U}(v|u)\cdot p^{\alpha-1}_{W|U}(w|u)}\right) \tag{2.61}$$

$$\mathrm{PRTE}_{Y->X}(k,l,m) = \mathrm{CRMI}(V,W|U) = \\ = \frac{1}{\alpha-1}\sum_{x_n}\cdots\sum_{x_{n-k+1}}\sum_{z_n}\cdots\sum_{z_{n-m+1}} p(x_n,\ldots,x_{n-k+1},z_n,\ldots,z_{n-m+1})\cdot \\ \cdot\log\sum_{x_{n+1}}\sum_{y_n}\cdots\sum_{y_{n-l+1}}\frac{p^{\alpha}(x_{n+1},y_n,\ldots,y_{n-l+1}|x_n,\ldots,x_{n-k+1},z_n,\ldots,z_{n-m+1})}{p^{\alpha-1}(x_{n+1}|x_n,\ldots,x_{n-k+1},z_n,\ldots,z_{n-m+1})\cdot} \\ \overline{\cdot p^{\alpha-1}(y_n,\ldots,y_{n-l+1}|x_n,\ldots,x_{n-k+1},z_n,\ldots,z_{n-m+1})}. \tag{2.62}$$

The results

We show that the newly introduced information-theoretic equation, named partial Rényi transfer entropy (PRTE), successfully detects the direction of information flow in an autoregressive model of order 1 [98]. We use the value of the order $\alpha = 3$. We compare the results produced by PRTE with other equations from Rényi's information theory, namely with the RMI, RTE and PRMI. We estimate the Rényi information-theoretic equations using a plug-in estimator. We first estimate the joint probability mass functions of the random vectors that appear in the analysis. We compute marginal probability mass functions from them, by summing out the extra random vectors. Then, we plug-in these values to the information-theoretic equations of interest. We estimate multivariate probability mass functions, of two, three and four dimensions, using the kernel density estimation toolbox for Matlab of Ihler and Mandel [52].

To make the calculations computationally feasible, we use the time lags equal to

$k = l = m = 1$, that is:

$$\mathrm{PRTE}_{Y->X}(1,1,1) = \frac{1}{\alpha-1}\sum_{x_n}\sum_{z_n} p(x_n,z_n)\cdot \cdot \log\left(\sum_{x_{n+1}}\sum_{y_n}\frac{p^{\alpha}(x_{n+1},y_n|x_n,z_n)}{p^{\alpha-1}(x_{n+1}|x_n,z_n)\cdot p^{\alpha-1}(y_n|x_n,z_n)}\right). \quad (2.63)$$

$$\mathrm{RMI}(X,Y) = \frac{1}{\alpha-1}\cdot\log\left(\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}\frac{p^{\alpha}_{XY}(x,y)}{p^{\alpha-1}_{X}(x)\cdot p^{\alpha-1}_{Y}(y)}\right). \quad (2.64)$$

$$\mathrm{RTE}_{Y->X}(1,1) = \frac{1}{\alpha-1}\sum_{x_n} p(x_n)\cdot\log\left(\sum_{x_{n+1}}\sum_{y_n}\frac{p^{\alpha}(x_{n+1},y_n|x_n)}{p^{\alpha-1}(x_{n+1}|x_n)\cdot p^{\alpha-1}(y_n|x_n)}\right). \quad (2.65)$$

$$\mathrm{PRMI}(X,Y|Z) = \frac{1}{\alpha-1}\cdot\sum_{z\in\mathcal{E}_Z} p_Z(z)\cdot\log\left(\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}\frac{p^{\alpha}_{XY|Z}(x,y|z)}{p^{\alpha-1}_{X|Z}(x|z)\cdot p^{\alpha-1}_{Y|Z}(y|z)}\right). \quad (2.66)$$

To make the estimation problem simpler, we transform the conditional probability mass functions into joint probability mass functions, according to 2.5: $p(x|y) = \frac{p(x,y)}{p(y)}$. Therefore, we have:

$$\mathrm{PRTE}_{Y->X}(1,1,1) = \frac{1}{\alpha-1}\sum_{x_n}\sum_{z_n} p(x_n,z_n)\cdot \cdot \log\left(\sum_{x_{n+1}}\sum_{y_n}\frac{p^{\alpha}(x_{n+1},y_n,x_n,z_n)\cdot p^{\alpha-2}(x_n,z_n)}{p^{\alpha-1}(x_{n+1},x_n,z_n)\cdot p^{\alpha-1}(y_n,x_n,z_n)}\right). \quad (2.67)$$

$$\mathrm{RMI}(X,Y) = \frac{1}{\alpha-1}\cdot\log\left(\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}\frac{p^{\alpha}_{XY}(x,y)}{p^{\alpha-1}_{X}(x)\cdot p^{\alpha-1}_{Y}(y)}\right). \quad (2.68)$$

$$\mathrm{RTE}_{Y->X}(1,1) = \frac{1}{\alpha-1}\sum_{x_n} p(x_n)\cdot\log\left(\sum_{x_{n+1}}\sum_{y_n}\frac{p^{\alpha}(x_{n+1},y_n,x_n)\cdot p^{\alpha-2}(x_n)}{p^{\alpha-1}(x_{n+1},x_n)\cdot p^{\alpha-1}(y_n,x_n)}\right). \quad (2.69)$$

$$\mathrm{PRMI}(X,Y|Z) = \frac{1}{\alpha-1}\cdot\sum_{z\in\mathcal{E}_Z} p_Z(z)\cdot\log\left(\sum_{x\in\mathcal{E}_X}\sum_{y\in\mathcal{E}_Y}\frac{p^{\alpha}_{XYZ}(x,y,z)\cdot p^{\alpha-2}(z)}{p^{\alpha-1}_{XZ}(x,z)\cdot p^{\alpha-1}_{YZ}(y,z)}\right). \tag{2.70}$$

For each of the four equations, we estimate the probability mass function of the highest dimension. For each of the marginal probability mass functions, we sum out the variables that are not of interest. That is, we estimate $p(x_{n+1},y_n,x_n,z_n)$, $p_{XY}(x,y)$, $p(x_{n+1},y_n,x_n)$ and $p_{XYZ}(x,y,z)$. Then, we have

$$\begin{aligned} p(x_{n+1},x_n,z_n) &= \sum_{y_n} p(x_{n+1},y_n,x_n,z_n)\\ p(y_n,x_n,z_n) &= \sum_{x_{n+1}} p(x_{n+1},y_n,x_n,z_n)\\ p(x_n,z_n) &= \sum_{x_{n+1}}\sum_{y_n} p(x_{n+1},y_n,x_n,z_n). \end{aligned} \tag{2.71}$$

$$p_X(x) = \sum_y p_{XY}(x,y) \text{ and } p_Y(y) = \sum_x p_{XY}(x,y). \tag{2.72}$$

$$\begin{aligned} p(x_{n+1},x_n) &= \sum_{y_n} p(x_{n+1},y_n,x_n)\\ p(y_n,x_n) &= \sum_{x_{n+1}} p(x_{n+1},y_n,x_n)\\ p(x_n) &= \sum_{x_{n+1}}\sum_{y_n} p(x_{n+1},y_n,x_n). \end{aligned} \tag{2.73}$$

$$\begin{aligned} p_{XZ}(x,z) &= \sum_y p_{XYZ}(x,y,z)\\ p_{YZ}(y,z) &= \sum_x p_{XYZ}(x,y,z)\\ p_Z(z) &= \sum_x\sum_y p_{XYZ}(x,y,z). \end{aligned} \tag{2.74}$$

The system under investigation is a stochastic autoregressive system of order 1, formed by three discrete coupled processes, X, Y and Z. The stochasticity is introduced by the presence of the Gaussian random noise, ϵ_1, ϵ_2, $\epsilon_3 \sim \mathcal{N}(0, 10^{-6})$:

$$\begin{cases} X[n] = 0.6 \cdot X[n-1] + \epsilon_1 \\ Y[n] = 0.9 \cdot Y[n-1] + X[n-1] + \epsilon_2 \\ Z[n] = 0.2 \cdot Z[n-1] + 0.5 \cdot Y[n-1] + X[n-1] + \epsilon_3. \end{cases} \tag{2.75}$$

The coefficients of the processes in the right side of the equations indicate the coupling strengths and the index of the processes show the coupling delay. That is, the process X is not coupled to any other process, it only receives input from its previous state. The process Y is coupled to the process X, with the strength of 1 and a coupling delay of 1. It also receives input from its previous state, with a coupling strength of 0.9. The process Z is connected to both processes and receives information from its previous state, with a strength of 0.2. The coupling direction is from Y and X to Z. The time delay is equal to 1 in both cases. The coupling strength is equal to 0.5, in the case of Y, and is equal to 1, in the case of X.

The system is initialized in a random state drawn from the uniform distribution, as $[x_0 \ y_0 \ z_0] = [1 + \mathcal{U}(0,1) \ 1 + \mathcal{U}(0,1) \ 1 + \mathcal{U}(0,1)]$, where $\mathcal{U}(0,1)$ represents the uniform distribution on the interval $[0 \ 1]$. We start the system in this random initial state and we run it forward in time for 50 time steps. Using these trajectories, we estimate the above mentioned Rényi information-theoretic equations. We average the results over 100 simulations. The PRMI and the PRTE correctly identify the coupling direction and the coupling delay between the processes Y and Z. The direction of the information flow is from Y to Z and the delay is equal to 1. The other measures, the PRMI and PRMI, are not able to identify either the direction or the delay of the information transfer in this system. These results indicate that the PRMI and the PRTE can extract structural information from the dynamical behaviour of a coupled system of stochastic processes [98]. They also show the need to extend the existing information-theoretic equations to new ones, as not all of them can provide accurate results for certain applications.

2.3 Kolmogorov complexity

Kolmogorov complexity or *algorithmic information theory* [[60], Ch. 14 of [25], [69]] represents a different paradigm of measuring information than Shannon's information theory and its generalizations. In the later case, objects are modeled as random variables. The uncertainty of an object is quantified as the average information of its probability distribution. In contrast, algorithmic information theory deals with the information contained in individual objects, instead of that contained in its probability distribution. Algorithmic information theory is useful when the objects cannot be easily modeled by random variables. For example, this would be the case of the executable model we are investigating. In Kolmogorov complexity, the information of an object is measured as the shortest binary program that can output the object on a universal computer. A universal Turing machine is such an example of a universal computer. In computability theory, the Turing machine represents an abstract model of computation, which is general enough to represent all computations performed by humans [Ch 1 of [69]]. A universal Turing machine represents a Turing machine that can simulate any other Turing machine. Excellent descriptions of algorithmic information theory and its applications can be found in [Ch 14 [25], [69]].

Unfortunately, the Kolmogorov complexity of an object is not computable. This fact can be explained through the halting problem in computability theory, proved by Alan Turing and Kurt Gödel [Ch 1 of [69], Ch 14 of [25]]. The halting problem refers to the fact that it is not possible to decide whether all programs will terminate or run forever, meaning that an algorithm to decide such a problem does not exist. The Kolmogorov complexity of a string represents the shortest program that can output the string. There is no possibility to find a minimal such program, from all possible programs, due to the halting problem. However, individual programs with a given input can be decided if they stop or not, but all possible programs cannot [Ch 14 of [25]]. Therefore, better and better programs to compress the string can be found, which are better and better approximations of the Kolmogorov complexity of the string. The Kolmogorov complexity $K(x)$ thus becomes the absolute lower bound of how much a string x can be compressed by a real-world compressor [68].

The normalized information distance (NID) introduced in [68] is based on the notion of the *information distance* [13], which is developed within the framework of Kolmogorov complexity [60]. As a consequence of the noncomputability of the

Kolmogorov complexity, the NID is noncomputable. The normalized compression distance (NCD) has been derived as a solution to the noncomputability problem of the NID [22].

We use the notation of [13] and [68]. Let x and y be two binary strings and x^* and y^* two binary programs that compute x and y, respectively. Let $K(x)$ be the Kolmogorov complexity or the algorithmic entropy of x. Let $K(x|y)$ be the conditional Kolmogorov complexity of x, given y. It is equal to the length of the shortest binary program that computes the string x, if the string y is also given as its input. The information distance [13] between the strings x and y is defined as

$$E(x, y) = \max\left(K(x|y), K(y|x)\right), \tag{2.76}$$

up to a logarithmic additive element.

The normalized information distance [68] is defined as

$$\mathrm{NID}(x, y) = \frac{\max\left(K(x|y^*), K(y|x^*)\right)}{\max\left(K(x), K(y)\right)}, \tag{2.77}$$

The information distance is an absolute measure of the difference between two objects, whereas the normalized information distance is a relative measure of the same quantity [68]. In classification, relative differences are necessary, instead of absolute ones, because objects of different lengths may share the same characteristics to belong to the same class, but would not be classified as such with absolute distances. They do not take into account the magnitude of the objects. Normalized distances are necessary when comparing objects at different scales.

Let C_x be the size of the compressed string x, C_y the size of the compressed string y and C_{xy} the size of the compressed string obtained by concatenating x and y. The normalized compression distance [22] is defined as

$$\mathrm{NCD}(x, y) = \frac{C_{xy} - \min\left(C_x, C_y\right)}{\max\left(C_x, C_y\right)}. \tag{2.78}$$

The NCD is a valid approximation of the NID by the mathematical theory developed in [22].

Chapter 3

Multidimensional scaling

3.1 Definition and properties of multidimensional scaling

Multidimensional scaling (MDS) is an exploratory analysis method. It facilitates the visual representation of high-dimensional data. The technique approximates high-dimensional dissimilarity scores, termed *proximities*, to two or three-dimensional Euclidean distances, so that the original objects can be represented in a two or three-dimensional figure. A matrix of dissimilarity scores between the high-dimensional objects is the input to the algorithm. It aims at representing these objects as points in a two or three-dimensional Euclidean space, such that the configuration of lower-dimensional points resembles as closely as possible the original configuration. This representation is not exact, as MDS is a technique to project high-dimensional data onto a lower-dimensional space. This operation results in some loss of information, measured by Kruskal's stress criterion Stress-1 [63], [64]. The raw stress has the formula

$$\sigma_r = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(d_{ij} - \hat{d_{ij}} \right)^2 \tag{3.1}$$

and the normalized stress

$$\sigma = \frac{\sum_{i=1}^{N} \sum_{j=i+1}^{N} \left(d_{ij} - \hat{d_{ij}} \right)^2}{\sum_{i=1}^{N} \sum_{j=i+1}^{N} d_{ij}^2}, \forall i, j = 1, N. \tag{3.2}$$

The variables in the MDS algorithm are: $\mathbf{X}$, the vector of points in the low-dimensional Euclidean space, $\mathbf{D}$, the matrix of Euclidean distances between the elements of $\mathbf{X}$, $\mathbf{P}$, the matrix of the proximities in the high-dimensional space, $\hat{\mathbf{D}}$, the matrix of distances in the low-dimensional Euclidean space, which approximates $\mathbf{D}$. The $\hat{\mathbf{D}}$ are the transformed $\mathbf{P}$. The MDS algorithm is the mapping between two spaces: the high-dimensional space, where similarities or dissimilarities between objects are defined, and the low-dimensional Euclidean space. The variables $\mathbf{X}$, $\mathbf{D}$ and $\hat{\mathbf{D}}$ characterize the Euclidean space and the $\mathbf{P}$ characterize the high-dimensional space. The MDS is an iterative algorithm that changes the configuration of points $\mathbf{X}$, such that $\mathbf{D}$ becomes closer to $\hat{\mathbf{D}}$, which are the transformed $\mathbf{P}$. The $\hat{\mathbf{D}}$ are related to the $\mathbf{P}$ either by a continuous function, or by having the same monotonicity. The stress criterion, σ, measures the error between the $\mathbf{D}$ and the $\hat{\mathbf{D}}$. The iterations stop, when the stress criterion, σ, is below a certain threshold. At this point, the final configuration of points is considered adequate enough to approximate the original proximities. The elements of $\mathbf{P}$ are referred to as p_{ij}, the elements of $\mathbf{D}$ are referred to as d_{ij} and the elements of $\hat{\mathbf{D}}$ are referred to as $\hat{d}_{ij}$.

There are two types of MDS, depending on how the proximities are transformed: *metric MDS* and *nonmetric MDS*, also named *ordinal MDS*. In metric MDS, the proximities are changed into the disparities by a continuous function. In nonmetric MDS, the algorithm retains only the rank of the proximities, not the actual values or a continuous transformation of them [Ch 9 [18]]. Since we have used nonmetric MDS in our analyses of the executable model, we will describe this version of MDS in more depth in the next section.

Nonmetric MDS

Nonmetric MDS is useful in situations where the proximity data cannot be explained be the continuous function f. It provides the best results when the rank order of the data is more informative than their actual values, such as in the case of our executable model. Here, the NCD data cannot be explained by an analytical relationship, i.e. a parametric model. The NCD between two system states is the result of compressing two files that contain symbols encoding a system's state, at a given time point. One NCD value is given by arithmetic operations on lengths of compressed files. Such type of experimental data does not have an analytical representation, such as a mathematical equation between the NCD values, which would allow us to compute

one NCD value from previous ones. Moreover, we are only interested in investigating how the states of different systems diverge or converge in time. Thus, our analyses only require the order of the proximity data. By using the nonmetric version of the MDS, we create points in a three dimensional Euclidean space, which have the same ordinal relationship as the original NCD data.

In nonmetric MDS, the disparities have the same ordinal relationship as the proximities, which means they have the same monotonicity, i.e

$$\text{If } p_{ij} \leq p_{kl}, \forall i,j,k,l = 1:N, \text{ then } \hat{d}_{ij} \leq \hat{d}_{kl}, \forall i,j,k,l = 1:N. \tag{3.3}$$

The nonmetric MDS algorithm performs the following steps:

- Start with an initial configuration of points: the solution of the metric MDS; $\Rightarrow d_{ij}^{(0)}$ can be computed for this iteration; compute the stress $\sigma^{(0)}$ for the starting configuration.
- $\mathbf{X}^{(k)}$, $d_{ij}^{(k)}$, $\hat{d}_{ij}^{(k)}$, $\sigma^{(k)}$ denote the $\mathbf{X}$, d_{ij}, $\hat{d}_{ij}$, σ, at the k^{th} iteration.
- Loop until the stress $\sigma^{(k)}$ is under the given threshold;
- At the k^{th} iteration, find $\hat{d}_{ij}^{(k)}$, with the property that they approximate $d_{ij}^{(k)}$, such that the residual sum of squares between the $\hat{d}_{ij}^{(k)}$ and the $d_{ij}^{(k)}$ is minimized. This is *a least squares minimization* problem. The $\hat{d}_{ij}^{(k)}$ have the same monotonicity as the p_{ij}. This problem is *a monotone regression* problem, which is solved using the *pool-adjacent-violators (PAV) algorithm* described below. At this step in the algorithm, the distances $d_{ij}^{(k)}$ are fixed and the variables are $\hat{d}_{ij}^{(k)}$.
- After the $\hat{d}_{ij}^{(k)}$ have been found, compute the stress $\sigma^{(k)}$ at the current iteration.
- Find a new configuration of points, $\mathbf{X^{(k+1)}}$. This is done using the *nonlinear conjugate descent method*. We refer the reader to the article [27], for more information on the properties of the nonlinear conjugate descent method.

In the MDS algorithm, the original proximities p_{ij} do not change their value. The only variables that change at each iteration are: the configuration of points $\mathbf{X}$ and, implicitly, the distances $\mathbf{D}$, the disparities $\hat{\mathbf{D}}$ and the stress criterion σ.

The nonmetric MDS has two important optimization steps:

- 1. At the end of the k^{th} iteration: find the next configuration of points $\mathbf{X^{(k+1)}}$, using the *nonlinear conjugate descent method*. This method represents an iterative optimization technique that finds the minimum of a nonlinear function. In our case, we want to minimize the stress criterion $\sigma^{(k)}$, which is a nonlinear function of the $\mathbf{D}^{(k)}$ and the $\hat{\mathbf{D}}^{(k)}$. The disparities $\hat{\mathbf{D}}^{(k)}$ are fixed and the stress is minimized with respect to the distances $\mathbf{D}^{(k)}$. After minimization, they become equal to $\mathbf{D}^{(k+1)}$, because the new configuration of points $\mathbf{X^{(k+1)}}$ is used at the $(k+1)^{th}$ iteration.

- 2. At the beginning of the $(k+1)^{th}$ iteration: after a new configuration of points has been found, minimize the stress criterion, to find new disparities. More explicitly, once we have found $\mathbf{X}^{(k+1)}$, which gives a new matrix of distances $\mathbf{D}^{(k+1)}$, we need to find a new matrix of disparities $\hat{\mathbf{D}}^{(k+1)}$, such that the new stress criterion $\sigma^{(k+1)}$ is minimized. The new disparities solve the least squares minimization problem, under the constraint that they have the same monotonicity as the original proximities. This is a monotone regression problem, which is solved by the PAV algorithm, described below. After this optimization step, we obtain a lower stress value $\sigma^{(k+1)} < \sigma^{(k)}$, because we do not use the old disparities $\hat{\mathbf{D}}^{(\mathbf{k})}$, which were optimally selected for the old set of points $\mathbf{X}^{(k)}$, but, because we have found a new vector of disparities $\hat{\mathbf{D}}^{(k+1)}$, that match the new configuration of points $\mathbf{X}^{(k+1)}$.

Monotone regression is the method of finding the minimum of a function, subject to inequality constraints. It is defined as [14]

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{i=1}^{n} w_i \cdot (y_i - x_i)^2 \\ \text{subject to} \quad & x_1 \leq x_2 \leq \ldots \leq x_n, \end{aligned} \tag{3.4}$$

where $w_i, \forall i = 1, \ldots, n$ are known weights and y_i are given.

The PAV algorithm, developed by several authors, [6], [110], [75], [64], [14] represents the most widely used algorithm to solve the monotone regression problem. The mathematical foundations of monotone regression and of the PAV algorithm can be found in [6], [110], [75], [14], while [64] is focused on its algorithmic approach, as an intermediate step in the problem of nonmetric multidimensional scaling. A recent review of monotone regression, together with different R implementations of some of its types can be found in [29].

To conduct our MDS analyses of the executable model, we used the built-in Matlab algorithm that solves the monotone regression problem with the PAV algorithm of [64]. To facilitate the computations of the PAV algorithm, we store the elements of $\mathbf{P}$ in a vector $\mathbf{p}$ of length $n = \frac{N^2-N}{2}$, such that it contains the upper triangle of $\mathbf{P}$, arranged in vector format. The vectors $\mathbf{d}$ and $\hat{\mathbf{d}}$ are defined similarly as $\mathbf{p}$. We note that the matrix $\mathbf{P}$ is symmetrical, so only the upper right triangle contains distinct values, which are used in the computations.

In our case, $\mathbf{x} = \hat{\mathbf{d}}$, $\mathbf{y} = \mathbf{d}$. The vector of weights $\mathbf{w}$ is optional. If it is not specified by the user, it will have all the elements equal to one. So, the monotone regression problem becomes

$$\begin{aligned} \underset{\hat{\mathbf{d}}}{\text{minimize}} \quad & \sum_{i=1}^{n} w_i \cdot \left(\mathbf{d}_i - \hat{\mathbf{d}}_i\right)^2 \\ \text{subject to} \quad & \hat{\mathbf{d}}_1 \leq \hat{\mathbf{d}}_2 \leq \ldots \leq \hat{\mathbf{d}}_n. \end{aligned} \tag{3.5}$$

The main elements of the PAV algorithm [64] are:

- We know the monotonicity of the disparities, which is equal to that of the proximities. But, as their actual values are unknown, we need to solve for them, provided that the distances are given.

- If the proximities $\mathbf{p}$ are dissimilarities, they are sorted in ascending order to ensure the monotonicity constraint. They are saved in the vector $\mathbf{ps}$ (if they are similarities the order is descending). We retain a vector of indeces, $\mathbf{is}$, to enable the recovery of the ordering of the original values from the results of the PAV algorithm. Then, $\mathbf{ps}_1 \leq \mathbf{ps}_2 \leq \ldots \leq \mathbf{ps}_n$, $\mathbf{ps}_i = \mathbf{p}_k$, and $\mathbf{is}_i = k$, $\forall i, k = 1 : n$.

- We use a new vector of distances, $\mathbf{ds}$, such that $\mathbf{ds}_l = \mathbf{d}_{\mathbf{is}_l}$, $\forall l = 1 : n$.

- The PAV algorithm uses three vectors, $\mathbf{ps}$, $\mathbf{is}$, $\mathbf{ds}$ and outputs the result in the vector of disparities, $\hat{\mathbf{ds}}$.

- Throughout the duration of the algorithm, the data remain separated into blocks. The algorithm starts with n blocks, each block containing only one value $\mathbf{ds}_i$, $\forall i = 1 : n$.

- Each block has a value associated with it. The value is computed as the average of the elements of the block $\mathbf{ds}_j$, $\forall j = 1 : Nb_k$, where Nb_k is the number of

elements of a certain block k. At the start of the algorithm, these averaged values are equal to $\mathbf{ds}_i$, $\forall i = 1 : n$. The partitioning of the blocks is performed using their associated values, computed as described above.

- The blocks change their number of elements, until the values associated with each block have the same monotonicity as that of the proximities. This change of partitioning takes place each time there is a violation in the monotonicity constraint of two adjacent blocks. In this case, these two blocks are merged and their corresponding value is computed again as the average of the elements of the newly-formed block.

- The algorithm stops when the vector of the associated values of the blocks have the same monotonicity as the vector of proximities $\mathbf{ps}$.

- Let k be a certain block from the final partitioning of the data $\mathbf{d}$, Nb_k be the number of elements of the block k and the indeces $\mathbf{is}_i$,...,$\mathbf{is}_j$, such that: their number is Nb_k and $\mathbf{d}_{\mathbf{is}_i}$,...,$\mathbf{d}_{\mathbf{is}_j}$ belong to the block k. At the end of the algorithm, each of the $\hat{\mathbf{ds}}_{\mathbf{is}_i}$,...,$\hat{\mathbf{ds}}_{\mathbf{is}_j}$ will be equal to the average of all the $\mathbf{d}_{\mathbf{is}_i}$,...,$\mathbf{d}_{\mathbf{is}_j}$.

Chapter 4

Discrete models of complex biological regulatory systems

4.1 A brief introduction to the immune system

In order to understand the elements involved in the biological model under study, we present a brief introduction to the immune system in vertebrates [91], for completness. The immune system represents an extremely complex interaction of elements that perform different roles to protect the organism against foreign microorganisms, such as bacteria and viruses. The skin is the first barrier out of multiple layers of defense, which prevents pathogens to enter the body. Other surface protection mechanisms are the mucosis of the digestive and the respiratory tracts. In addition, the vertebrate organism has a series of intricate internal defense mechanisms. The following four are the most important types of internal defenses to destroy foreign microorganisms that have entered the body.

Cells that destroy pathogens. The most important of the cells that destroy pathogens are *the macrophages*, *the neutrophils* and *the natural killer cells*. These cells run through the body and destroy any pathogens they encounter. *The macrophages* are large cells that destroy bacteria and viruses, by engulfing them. They can ingest one microbe at a time and perform this action several times. The process of assimilating and destroying a microorganisms is termed *phagocytosis* and the cells that perform this action are called *phagocytes*. *The neutrofils* are cells that destroy multiple bacteria at one time by chemical means, but in this process they destroy themselves too. *The natural killer cells* identify and destroy the cells of the

body that have been infected by pathogens. They can also destroy cancer cells. All of these cells of the immune system have the capacity to distinguish between the body's own cells and foreign particles. In autoimmune diseases, this capacity fails and the immune system starts destroying the body's own cells.

The complement system. The complement system represents a chemical protection system made up of approximately 20 proteins that enhance the immune defense of the cells of the immune system.

The inflammatory response. Infected cells release chemicals that promote the dilation of the blood vessels around an infected site. This results in increased blood flow to these areas, such that the macrophages and neutrofils can reach the site of infection faster. These processes cause the redness and swelling of the infected areas.

The temperature response. The temperature response is triggered when macrophages that have encountered a pathogen release a chemical called *interleukin-1.* This chemical, together with toxins produced by bacteria, make the hypothalamus elevate the body's temperature, causing fever. This enhances the process of phagocytosis and reduces the levels of iron in the blood, required for bacteria to multiply.

These four major elements are termed *general or nonspecific response mechanisms*, because they are not designed for a specific type of pathogen, but act on any microorganism that has entered the body and has the potential to cause harm. The next layer of defense mechanisms is named the *specific immune response.* This even more elaborate layer has defense procedures customized to very specific pathogens and it can remember microorganisms encountered in the past. In this way, the immune system's response to previously encountered pathogens is faster and more efficient. In response to the presence of foreign molecules called *antigens*, the specific immune response creates special proteins, named *antibodies.* An antigen represents a molecule that does not belong to the body, for example a molecule found on the surface of bacteria. Each antibody is specific to a type of antigen. This specificity enables the immune system to remember pathogens and to mount a faster immune response when they are found at a later time.

4.1.1 The types of cells of the immune system

The immune system is a heterogeneous, multi-layered system, formed by a multitude of types of cells and chemicals that work together to defend the body against

pathogens. It represents a cooperation of these cells and chemicals, which makes it a perfect example of a complex system.

Leukocytes, also called white blood cells, are the most important cells of the immune system. They are found in the blood, lymph nodes, spleen and liver. Leukocytes develop in the bone marrow from the same type of cells, the hemopoietic stem cells, as erythrocytes. The function of leukocytes, or white blood cells, is in the immune respunse and the function of erythrocytes, or red blood cells, is to transport oxigen and carbon dioxide. The most important types of leukocytes are: *neutrophils*, *eosinophils*, *basophils*, *monocytes* and *lymphocytes*. *Neutrophils* are produced in the largest amount of all the leukocytes and they are the first to respond in an immune reaction. *Eosinophils* respond to parasites and are present in allergies. *Basophils* play a role in the inflammatory response. *Monocytes* change into macrophages at the site of infection. *Lymphocytes* are part of the specific immune response and they secrete antibodies. The lymphocytes are divided into *T cells* and *B cells*. The lymphocytes cannot perform phagocytosis. The other types of leukocytes (neutrophils, eosinophils, basophils and monocytes) can complete phagocytosis.

The T cells. T cells are produced in the bone marrow and migrate to the thymus, where they mature. Each T cell can recognize one antigen. T cells are divided into four major categories: *helper T cell*, *inducer T cell*, *cytotoxic T cell* and *supressor T cell*. *Helper T cell* represents the primary immune response cell that starts the immune defense. *Inducer T cell* helps the development of the T cells in the thymus. *Cytotoxic T cell* destroys infected cells and *supressor T cell* ends the immune response, after the infection has been cleared.

The B cells. B cells become mature in the bone marrow and then travel through the blood and the lymph. Each B cell is specialized to detect one antigen. When a B cell finds its antigen, it divides into plasma cells. Each plasma cell that results is a cell that produces antibodies for the specific antigen, which was the target of its original B cell. These antibodies are released and travel through the body attaching themselves to the pathogens that have this specific antigen. As a result, other B cells can recognize these pathogens and destroy them.

Other chemicals of the immune system. *Interferons* are proteins secreted by cells that have been infected with viruses. The macrophages and the natural killer cells respond to these proteins and come to the infected site to destroy the pathogen. Macrophages that are dealing with a viral infection start producing *cytokines*, for example $\gamma-$interferon, which are molecules that act on monocytes to turn them

into macrophages, activate helper T cells and raise the temperature of the body. The macrophages that have encountered a pathogen produce interleukin−1, which activate the helper T-cells. The helper T-cells are crucial for the immune response. They activate all other major defensive cells: inducer T-cells, cytotoxic T cells, suppressor T cells and B cells. Helper T cells start the two types of immune response: *cell-mediated immune response* and *humoral immune response*. The Cytotoxic T-Lymphocyte Antigen−4 (CTLA-4) is a protein found on the surface of activated T cells, with a role of inhibiting the function of T cells [54]. It is a protein that spans the entire wall of the cell and is found on the surface of the cell, that is, it represents a transmembrane protein. It inhibits the proliferation of other T cells. Interleukin−10 (IL-10) is an inhibitory cytokine and CTLA-4 is an inhibitory receptor [111].

The cell mediated immune response represents the elimination of own cells that have been infected by viruses or have become abnormal for other reasons, such as in cancer. Helper T-cells become activated by interleukin−1, which is secreted by macrophages. Such cells are called *activated helper T-cells*. The main actions of the activated helper T cells are: *proliferation*, *activation*, *induction* and *supression*. *Proliferation* represents the action of making other T cells divide. *Activation* represents the process of bringing more macrophages to the infection site. Through *induction* they activate the inducer T cells. By *supression*, they activate suppressor T cells, which shut down the immune response after the infection has been cleared. In this type of immune response, other actions of the cells include the destruction of virus infected cells by *cytotoxic T cells* and the development of *memory T cells*. They remember the antigens they have destroyed, for a faster response to future infections with the same antigens.

The humoral immune response constitutes the functioning of the B cells in protecting the body. B cells do not have the capacity to destroy infected cells, but they can create markers for specific antigens. These markers are named *antibodies*. They facilitate the recognition of their corresponding antigens, which can be destroyed by the general or nonspecific immune defense mechanisms. This type of immune response is also triggered by the helper T cells. When a B cell has encountered its specific antigen, the cell binds to it. The activated helper T cells cause the B cell to proliferate. This proliferating B cell differentiates into plasma cells. *Proliferation* represents the increase in numbers of cells by frequent cell division [67]. *Differentiation* represents the increased specialization in function and structure of cells that came from cells with no specific function [67]. Unspecialized cells multiply and then

divide into specialized cells, which have a particular function that the old cells could not accomplish. Some B cells do not differentiate into plasma cells. These cells are termed *memory B cells* and they have the same function as the memory T cells. The humoral immune response is also suppressed after the infection has been stopped.

4.2 Executable model of the regulation of cytokines within the human T-cells

4.2.1 The biological model

The biological model under investigation was developed by [114]. A detailed description of how the biological experiments were performed can be found there. We will restrict our description to the main elements of the model and how they influence each other. We will skip the biological details of how the experiments were performed. These are beyond the scope of this study.

The biological model of the functions of regulatory T cells of the human immune system is an illustration of the effect of *the heat shock protein 60* (*HSP60*) on a population of *the regulatory T cells* of the human immune system. Heat shock proteins are a family of proteins that are produced by cells under stress. Stress refers to either changes in the temperature surrounding the cell, infections or any other type of external actions that would aim at destroying the cell. The traditional roll of heat shock proteins is as molecular chaperones in the correct functioning of the cell and in the correct folding, assembly and transportation of proteins [70]. Molecular chaperones are proteins that assist in the forming of other molecular structures, such as proteins or cells, but are not part of their normal functioning [50]. Heat shock proteins are one type of molecular chaperones [50]. More recently heat shock proteins have been found to play a crucial role in the immune response [112], [108].

Regulatory T cells and helper T cells are two categories of $CD4^+$ T cells. Regulatory T cells are also known as suppressive T cells, because of their role to end the immune response and to control it against the self. Dysfunction of the regulatory T cells has been linked to autoimmune diseases. The extremely complex function and regulation of these subtypes of T cells and their role in the immune response and in autoimmune diseases are not completely understood, which makes them the subject of active research in the field of immunology [96], [24], [111].

$CD4^+CD25^+$ T cells are *regulatory T cells* that develop in the thymus. *$CD4^+CD25^-$*

T cells are *naïve T cells* that can be found in the periphery, that is, the blood or other parts of the body, where an immune reaction has been triggered. The immunosuppressive function of the regulatory T cells is better understood, than whether they are produced in a specific organ, such as the thymus, or if they can be induced from other types of T cells throughout the body [21]. The aim is to identify how these regulatory T cells are generated from other T cells and what molecules facilitate this transformation [21]. The extensive recent review by Brenu et.al. [20] provides further details on the relationship between heat shock protein and the immune system, in particular regulatory T cells.

Detailed description of the function of T cells. We present in the following a more detailed description of the types and the functions of T cells [54], to better understand how the different molecules and cells of [114] influence each other. When a pathogen is recognized by cells of the immune system, these cells produce cytokines and chemokines that trigger the immune response to the type of pathogen that has been recognized. The role of the innate or general immune response is to eliminate an infection or to not let it spread until the adaptive or specific immune response is set up, as a reaction to the particular pathogen that caused the infection. Since it is directed at a certain pathogen, the adaptive response is much more precise than the general response. However, the adaptive one is much more complex and takes more time to set up.

Naïve T cells are mature cells that have not found their particular antigen, while circulating through the body [Ch 8 [54]]. When they encounter their specific antigen, they have to be activated into a particular type of T cell to destroy that pathogen, which the antigen came from. Activation refers to the process of these cells being made to *proliferate* and *differentiate* into a particular type of T cell. When they encounter their specific antigen, the naive T cells multiply and then differentiate into effector T cells. These are the cells that destroy the pathogen, which the antigen came from. T cells that have multiplied and differentiated create effector T cells that are of different types and can either kill infected cells or activate other responses of the immune system [Ch 1 [54]]. Cytotoxic T cells are also called CD8 T cells, by the CD8 molecule found on their surface. CD4 T cells are a class of T cells called helper T cells. CD4 T cells are further classified into Th1 cells and Th2 cells. Th1 cells destroy bacteria found in a cell and Th2 cells activate B cells, which, when encountering its antigen, become activated with the help of Th2 cells to produce antibodies.

The description of how the biological model works

$CD4^+CD25^-$ are effector T cells and $CD4^+CD25^+$ are regulatory T cells [114]. In the biological model of [114], regulatory T cells (Tregs) suppress the proliferation of $CD4^+CD25^-$ T cells. They found that HSP60 applied to Tregs has a greater effect to downregulate $CD4^+CD25^-$ T cells and $CD8^+$ T cells, than in the case when HSP60 is not present. HSP60 acts on $CD4^+CD25^+$ Tregs through the TLR2 receptor. TLR stands for Toll-like receptors, which are cell-surface membrane proteins that are present on the surface of cells of the immune system [Ch 2 [54]]. They recognize particles from pathogens and initiate immune responses as a result. Activation of $TLR2$ leads to the activation of the transcription factor NF-κB, which induces genes responsible for producing cytokines, chemokines and other molecules involved in the immune response [Ch 2 [54]]. The heat shock protein HSP60 determines the Tregs to increase the phosphorylation of the AKT, Pyk2 and p38 and to turn off the phosphorylation of ERK. *Phosphorylation* [67] refers to the action of adding a phosphate group to a protein, thus changing its function. The Tregs increase the production of IL-10 and TGF-β and suppress the activated $CD4^+CD25^-$ T cells, which were activated by TCR activation. As a result, these effector T cells downregulate ERK, NF-κB and T-bet. This leads to a decrease in their proliferation and to a decrease in the secretion of the proinflamatory cytokines IFN-γ and TNF-α. In addition, the production of IL-10 increases.

In the biological model under study [114], [95], the molecule aCD3 stands for anti-CD3, which is an antibody, that is, a molecule designed to activate T cells. It binds the CD3 protein complex of the TCR, on the surface of T cells, thus activating them [Ch 6 of [54]]. The two activators of this model, HSP60 and aCD3, are present in the environment from external sources, in the beginning of the experiment. They are not replenished throughout the simulation. Thus, they diffuse in the environment and disappear towards the end of the experiment. The major biological events that take place in this system are the following: the populations of Tregs and naïve T cells are activated by the aCD3. In addition, HSP60 co-activates the Treg population of cells. As can be seen in the figure 1 of [95], the Tregs differentiate into activated Tregs, which produce the cytokines CTLA-4 and IL-10 upon activation. The naïve T cells differentiate into Th1 helper T cells, which proliferate into activated Th1 helper T cells. They secrete the inflammatory cytokine IFN-γ, which activates them further, in a positive feedback loop. The cytokines CTLA-4 and IL-10 supress the proliferation

of the activated Th1 helper T cells and the secretion of the cytokine IFN-γ. Through IL-10, a part of the activated Th1 population of cells differentiates into suppressed Th1 cells, which secrete the IL-10 cytokine. As a result, the concentration of IL-10 increases in the environment.

4.2.2 The computational model

GemCell [5] is an example of a generic executable mode, designed as a tool for experimental biologists. Its name stands for *Generic Executable Modeling of Cells*. *Generic* signifies that the program is not fixed for a particular biological model, but that it can accommodate a large class of systems that share a few fundamental characteristics. *Executable modeling* refers to the modeling paradigm selected to describe biological systems seen as reactive systems. We have explained this choice of modeling biological systems, in more depth, in the introductory section 1.3. The term *Cells* indicates that the cell is the building block of the model.

The executable model we are investigating in this chapter is created by customizing the generic model GemCell to a particular biological system, namely to the regulation of cytokines within the human immune system [114], which was described in detail in the previous section. GemCell is a generic model that has basic biological rules and a database, which contains the biological information about the specific system. The purpose of the program is to model different types of biological systems, by changing the database with details specific to each system. The general rules are identical for a broad class of systems. The basic building block of the model is the cell. The generic dynamical laws of the cell's behaviour are: *proliferation*, *movement*, *death*, *secretion of molecules* and *reception of external signals*. For other approaches in modeling and analysing T cell behaviour in the immune system, we refer the reader to the extensive review of [78].

The GemCell program is composed of four elements: the statechart formalism [48] of the generic dynamical cell laws, a database with biological specifics, the linkage of the statechart model with the biological database and the visualization of the output of the model. The statechart formalism [48] is a visual modeling language designed for reactive systems, with states and transitions between states, to account for all the complexities of such systems. The output of GemCell can be in the form of a text file or as a two-dimensional Matlab figure, for each time point of the simulation. One figure shows the environment as a two-dimensional grid and, on each grid location,

the cells and the molecules. The colour of each cell indicates its type. The area of the grid where molecules are present is shaded with gradients of colours, to show how their concentrations vary across the grid. In addition, the executable model can produce figures displaying numbers of cells and concentrations of molecules over time.

One execution of the model refers to starting the model in an initial state and running the model forward in time, for 31 time points. From one time step to another, the changes in the state of the system take effect synchronously. The time points denote hours of real time and the difference between two consecutive time points is equal to one hour of real time. At each time step, the model outputs the state of the system in the form of a text file. The state of the system is encoded as numerical and alphabetical symbols, which describe the different elements of the system and the events that take place. The environment is modeled as a two-dimensional grid of size 20×20. Each line of the state file contains the information about the cells and molecules found on a given grid location, at the time point for which the file has been created. The following information is encoded for the cells: the name, the index in the database, the remaining life span, the dynamical state, which is one of the generic dynamical laws of the cell's behaviour, the name of the expressed receptors and their expression levels. In the case of the molecules, the file encodes their name and their concentration.

The initial number of cells is 100, which are divided into two populations and whose location on the grid is random. The percentage of the cells that belong to each populations can be adjusted according to the experiment. In addition, different molecules can be eliminated from the experiments, partially or totally. We analysed the output files of the executable model in the following conditions:

- the wild type system, which has the initial ratio of cells as 10 regulatory T cells (Tregs) to 90 naïve T cells (nTh);
- the knock-out perturbations at 100% efficiency, where the molecules IL-10, IFN-γ and CTLA-4 are each eliminated from the system;
- perturbed systems with IL-10 knock-out at 25% efficiency, 50% efficiency, 75% efficiency and 100% efficiency;
- a random setup, where the numerical values of the parameters are random, but the general biological rules are the same as for the wild type;

- systems similar to the wild type but with different ratios of the initial populations of cells, namely 30 Tregs to 70 nTh, 50 Tregs to 50 nTh , 70 Tregs to 30 nTh, 90 Tregs to 10 nTh;
- two systems: one that contains only the molecular features of the wild type and one that contains only the cellular features of the wild type.

A knock-out perturbation of a system refers to the action of completely eliminating a target molecule from the system. We refer to this phenomenon as a knock-out at 100% efficiency [95]. However, in biological experiments, it is not always possible to totally eliminate a molecule for the experiment. Even if the molecule is knocked-out, it can still be present at lower concentrations than before this action. In this case, we refer to the phenomenon as a partial knock-out or as a knock-out at an efficiency lower than 100%. Therefore, it is of great experimental interest to quantify the effect of partial knock-outs of the molecules of interest, on the overall dynamical behaviour of the entire system. To this end, we develop methods to predict the amount of the efficiency of a knock-out required to have the desired effect on the system's dynamics. This case is a perfect example of the need to understand and quantify the structure-dynamics relationships in complex biological systems.

4.3 The NCD analysis of the executable model

Motivation

For each system, we wanted to see how the state of the system changes in time. In this way, we would be able to detect important events that take place in the execution of the model. For example, we would be able to see how the system's state changes when a biological event happens. For example, a new molecule is being secreted, or an old population of cells gets transformed into another population of cells. We would also like to quantify the magnitude of the event, by which we mean if the NCD changes significantly due to a biological process that changed the state of the system significantly. We want to know if the NCD can detect how much the system's state has changed, when biological processes take place. To this end, we computed the NCD between two consecutive states and displayed it over time, i.e.

$$[\mathrm{NCD}(t), \mathrm{NCD}(t+1)], \forall t \in \{0, \cdots 29\}. \tag{4.1}$$

NCD analysis of the executable model

We compute the NCD using the xz Utils compressor. We chose this compressor out of several other compressors, such as 7zip, bzip2, gzip, because it provides the widest range of the NCD values. It provided us with values between $[0.64 \times 0.94]$. The encoding of a state of the system is complex, because it has a large number of state variables that change their value at each time point, they are of different type and they can perform a wide variety of actions. As a result, in the beginning of the simulation, when there are not many cells and molecules in the system, due to the complexity of the model, the NCD will have values around $0.7 - 0.8$. These values correspond to increased similarity. When biological events change the state of the system in a significant way, the NCD values increase to $0.94 - 0.95$, indicating that two consecutive states are very different from one another. The result of the NCD analysis is displayed in the figure 2 of [95]. The NCD measure can detect when major biological events take place and affect the state of the wild type system. In addition, we apply the NCD measure to the state of the system divided into molecular information and cellular information. The results are shown in the figure 4.1. The figure 4.1 was published as the supplemental figure 1 of [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`. We

found that the molecular features of the system had a greater influence over the overall dynamical behaviour of the wild type, than the cellular features did.

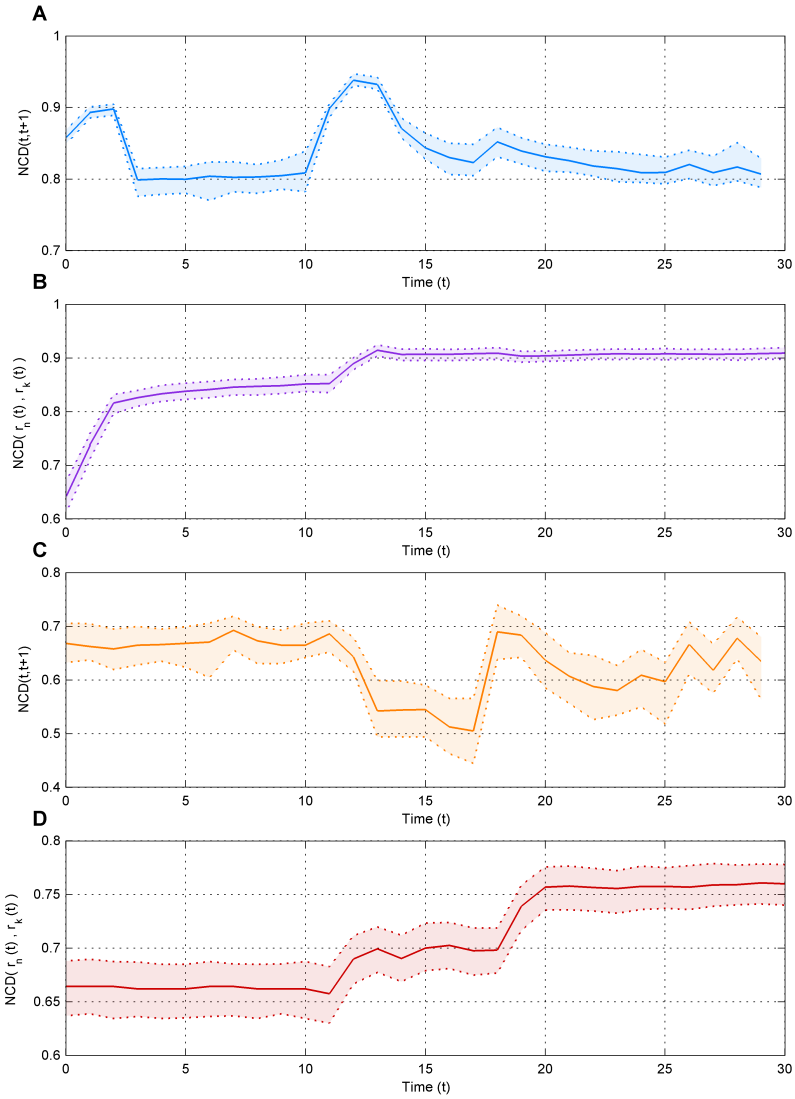

Figure 4.1: This figure was published as the supplemental figure 1 in [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`. The NCD applied to the molecular information and to the cellular information, separately.

4.4 MDS analysis of the executable model

Motivation

When the aim is to classify different structural perturbations of the wild type system, from the dynamical behaviour, the representation of the NCD versus time of each individual system does not clearly reveal the differences between them. The reason is the complex encoding of system states, which results in the NCD values being grouped together in a small interval close to 1. In order to solve this problem, we enhanced the NCD comparison of the dissimilarity of system states with three-dimensional multidimensional scaling (MDS). In other studies, two dimensional MDS has been used with an NCD dissimilarity matrix for classification purposes: ordered, critical and chaotic random Boolean networks were separated into distinct dynamical regimes, with a combination of a two-dimensional MDS representation of the NCD data between their trajectories [84]. A two-dimensional MDS visualization of the NCD dissimilarity matrix of mitochondrial genomes of 24 mammalian species was used in classification tasks to test hypotheses in mammalian evolution [22].

MDS analysis of the executable model

Our purpose is to analyze, at a global level, models of biological systems that integrate diverse information from multiple scales. The problem we are aiming to solve with MDS is to detect biological perturbations in an executable model of the human immune system. Our MDS analysis methods provide a systems' level description of how the executable model behaves dynamically, under different structural changes. The biological conditions under analysis are presented in the section 4.2.2, regarding the computational model.

The NCD displayed over time problem. The visualizations of the NCD over time, for each of the four systems separately, does not reveal any significant differences between the systems. The reason for this phenomenon is the large values of the NCD, which are concentrated in a small interval close to 1, $[0.8 \times 0.95]$. After a few major biological events have taken place, the state of the system becomes extremely complex, due to the presence of a wide variety of cells and molecules, together with their actions and states. After the time point of the last significant event in the execution of the model, the NCD values remain very large, in the interval $0.9 - 0.95$. They lie in this interval because the states of the systems have become

extremely complex and the files that describe these states encode a great amount of information. However, if we combine the entire NCD information we have obtained from the four setups, in a global dissimilarity matrix, we can visualize their dynamical behaviour in the same figure, with three-dimensional nonmetric MDS. This MDS representation enhances the subtle differences between the four systems, which were not identifiable in the displays of the NCD over time.

The MDS solution. By a trajectory, we denote the evolution of a quantity in time. In our case, the quantity we are observing is the similarity of two consecutive states. The states can be from the same system or from different systems. This similarity is measured by the NCD, which is the proximity measure for the MDS analysis. Our objective is to visualize the trajectories of similarity of states of the four setups, in one MDS figure. To this end, we employ the nonmetric version of the MDS algorithm and we create the NCD dissimilarity matrix for its input. We choose three-dimensional nonmetric MDS, instead of the two-dimensional version, because the trajectories are more clearly separated in the first case.

In the MDS analyses, we display the mean trajectories of the systems, averaged over 50 simulations of the executable model. One simulation or run entails executing the model for 30 time points from an initial random state. This means starting the model with 100 cells placed at random on the grid (at time point $t = 0$) and generating the states of the system for $t = 1 : 30$ time steps. We create the NCD dissimilarity matrix using all the runs, which places all the corresponding trajectories in the MDS figure. After this step, we compute the mean value of the trajectories for display and comparison. We denote the NCD dissimilarity matrix as $\mathbf{D_{NCD}}$. Each element of $\mathbf{D_{NCD}}$ represents the NCD value between two system states. The states can belong to the same system or to two different systems. In the first case, the states can be part of the same run or of distinct runs. In either of the two cases, the time point of the states can be identical or distinct. Let N_s denote the number of systems under the MDS analysis, S_i denote the i^{th} system, $\forall i = 1 : N_s$, r_j denote the j^{th} run, $\forall j = 1 : 50$, t_k the time point, $\forall k = 0 : 30$. Then,

$$\begin{aligned}\mathbf{D_{NCD}}(l, m) = \mathrm{NCD}(S_{i_1}(r_{j_1}(t_{k_1})), S_{i_2}(r_{j_2}(t_{k_2}))),\\ \forall i_1, i_2 = 1 : N_s, \forall j_1, j_2 = 1 : 50, \forall k_1, k_2 = 0 : 30.\end{aligned} \tag{4.2}$$

We performed the MDS analysis for the following experimental conditions:

1. In the case of $N_s = 4$ setups (the wild type, IL-10, IFN-γ and CTLA-4, all the perturbations at 100% efficiency), for each of the systems, we have 50 runs $\times$

30 time points = 1550 states. So, $l, m = 1 : (4 \times 1550) = 6200$ and the size of the $\mathbf{D_{NCD}}$ will be 6200×6200;

2. In the case of $N_s = 5$ setups (the wild type and IL-10 at the following efficiencies: 25%, 50%, 75% and 100%), for each of the systems, we have 50 runs $\times$ 30 time points = 1550 states. So, $l, m = 1 : (5 \times 1550) = 7750$ and the size of the $\mathbf{D_{NCD}}$ will be 7750×7750;

3. In the case of $N_s = 5$ setups (the wild type and similar systems, with different ratios of the initial populations of cells: 30 Tregs to 70 nTh, 50 Tregs to 50 nTh , 70 Tregs to 30 nTh, 90 Tregs to 10 nTh), for each of the systems, we have 50 runs $\times$ 30 time points = 1550 states. So, $l, m = 1 : (5 \times 1550) = 7750$ and the size of the $\mathbf{D_{NCD}}$ will be 7750×7750.

The nonmetric MDS results. In the following paragraphs, we will describe the results of our NCD+MDS analyses as presented in [95]. For the biological condition 1 described above, we show the results in the figure 4.2. The figure 4.2 was published as the figure 3 of [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`. The four trajectories start from the same point and are approximately identical for a few time points. Then, as major biological events take place, the trajectories of the systems start to diverge. This means that our information-theoretic measures can reveal in which way each perturbed system behaves differently, not only that all the knock-out systems are different from the wild type, but how each of them differs from the others.

To show a more quantitative illustration of how these trajectories diverge, we computed the Euclidean distance between the MDS coordinates of the wild type trajectory and each one of the perturbations, for all the time points. We can see that the system with the CTLA-4 knock-out has the most similar dynamics to that of the wild type, followed by IL-10, and lastly, by IFN-γ. We can also notice that IL-10 and CTLA-4 knock-out systems have a similar shape of the trajectory. These two systems react similarly to the major biological events that take place. But, IL-10 is significantly more different from the wild type, than CTLA-4 is. As time progresses, the behaviour of IFN-γ increases in dissimilarity from that of the other systems. These findings are in agreement with the biological roles of the cytokines. The cytokines IL-10 and CTLA-4 are secreted by the same population of cells, the

activated Tregs, and have an inhibitory role on the other population of T cells, the activated $Th1$ helper cells. In contrast, the cytokine IFN-γ is secreted by the activated $Th1$ helper cells and has an inflammatory role, by activating further this population of cells.

For the biological condition 2 described above, we show the results in the figure 4.3. The figure 4.3 was published as the figure 5 of [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`. As in the previous case, the trajectories are similar for the first few time points, until the point when the two populations of cells proliferate, differentiate into new types and the secretion of new molecules takes place. After that point, the trajectories diverge significantly for the higher efficiency cases, while the behaviour of the 25% perturbed system is approximately identical to that of the wild type. A knock-out efficiency of over 50% is necessary to have any significant effect on the dynamical behaviour. A linear increase in the efficiency of the knock-out of IL-10 produces a nonlinear effect on the dynamics of the perturbed system, compared to that of the wild type.

For the biological condition 3 described above, we show the results in the figure 4.4. The figure 4.4 was published as the supplemental figure 2 of [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`,
`http://journals.plos.org/plosone/s/content-license`.
As the ratio of the two populations of cells is increasingly different from that of the wild type of 10 Tregs to 90 nTh, up to 90 Tregs to 10 nTh, the trajectories are increasingly more divergent. However, the shapes of the trajectories are similar to each other and to that of the wild type. This indicates that the systems respond dynamically in a similar manner to the changes that take place.

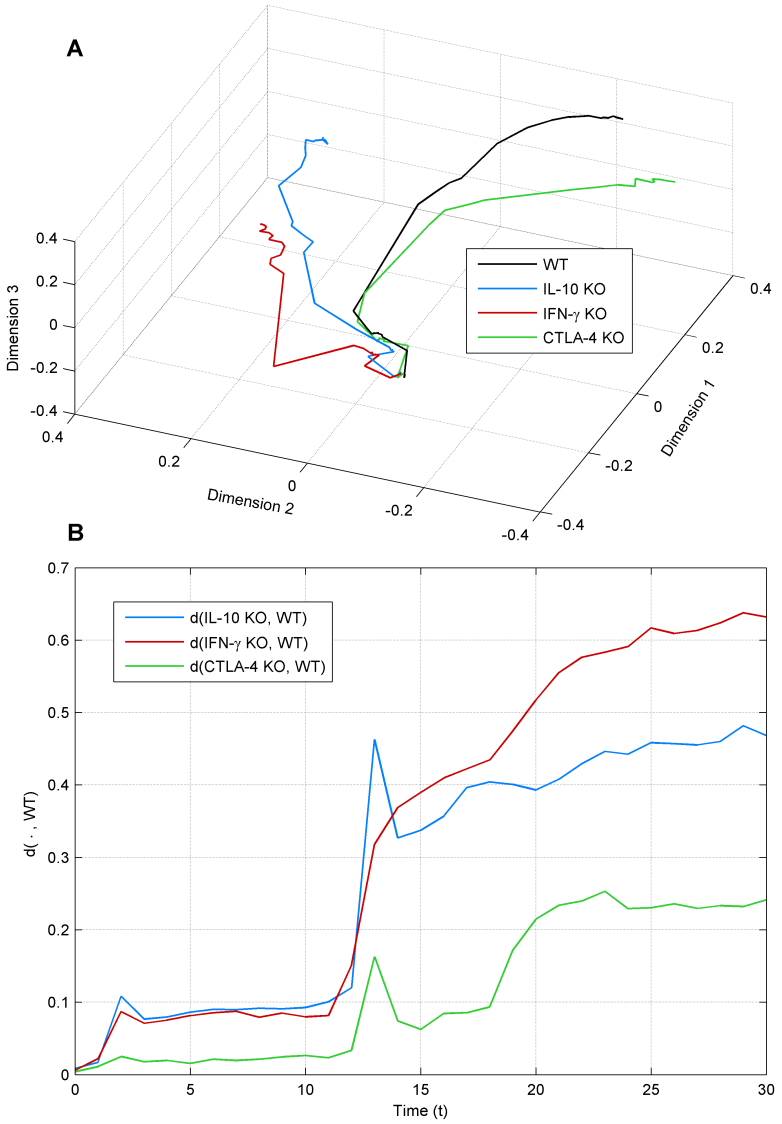

Figure 4.2: This figure was published as the figure 3 in [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`. The MDS representation in three dimensions of the wild type, the IL-10, IFN-γ and CTLA-4, all the perturbations at 100% efficiency.

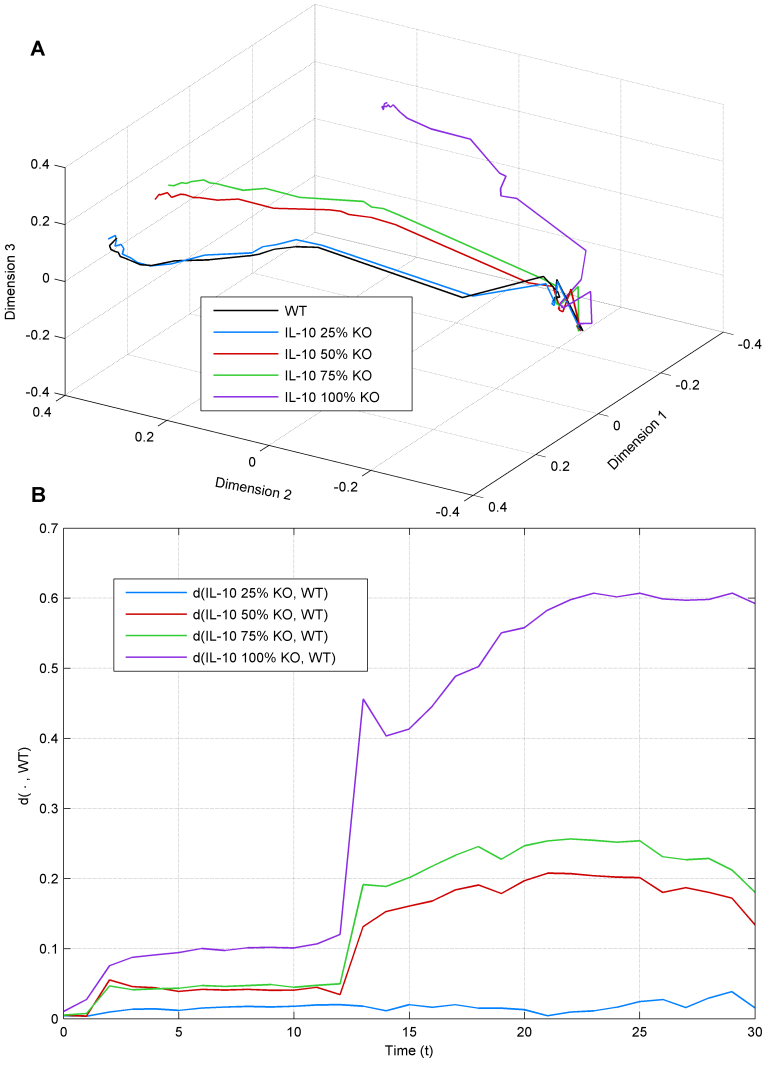

Figure 4.3: This figure was published as the figure 5 in [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`. The MDS representation in three dimensions of the wild type and of several partial knock-outs of the IL-10.

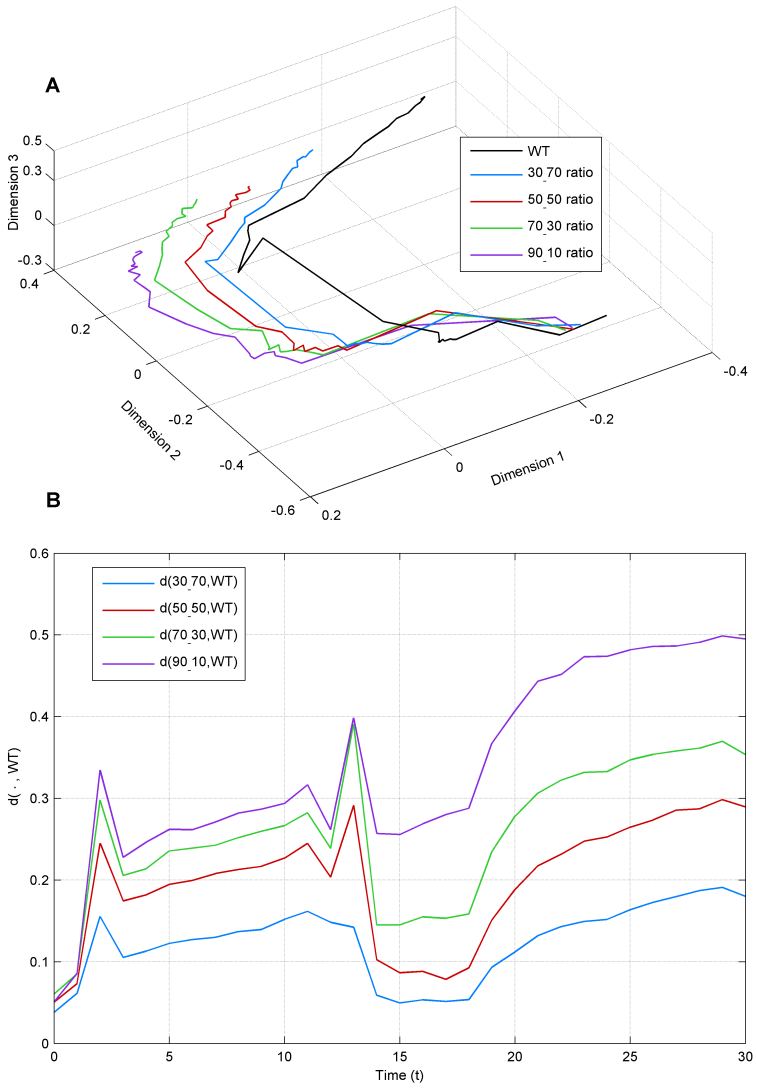

Figure 4.4: This figure was published as the supplemental figure 2 in [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`. The MDS representation in three dimensions of several ratios of the initial two populations of cells of the wild type.

4.5 Probabilistic convergence maps

The MDS analysis has provided a comparison of the evolution in time of various setups of the biological model. We develop an additional analysis method, of statistical nature, where time is eliminated. These two types of analysis complement each other to give a more thorough insight into the structure-dynamics relationships of the biological system. They validate each other, because, with two conceptually different procedures, we obtain the same conclusions.

The probability mass function of the NCD points

We take a different approach from the MDS visualization methods that have been described so far. We develop statistical characterizations of the dynamical behaviour of the executable model. They can be interpreted as classification procedures of different perturbations of the wild type system. Here, we provide statistical descriptions of how the similarity of two states of a system changes from one time point to the next. The statistical displays of the figure 4 of [95] show the probability that a given degree of similarity between two states, turns into another degree of similarity between the same states, at the next time point. Thus, these figures illustrate the convergence and divergence of the states of a system, in probabilistic terms. We use a probabilistic approach to describe the similarity of states, because the executable model is stochastic, that is, it accounts for the randomness in the biological system that has been modelled.

In addition to the systems investigated in the previous section, we add a random model to the analysis. We compute the probability mass function of the NCD points for each of these five systems, using two-dimensional kernel density estimation [19]. We create three-dimensional probabilistic descriptions for each individual system and we display them for comparison as two-dimensional contour plots. Firstly, we report how we computed the NCD data, then we describe the estimation procedure of the probability mass functions and we illustrate how we obtained the contour visualizations.

For this probabilistic analysis, unlike the nonmetric MDS visualizations, the NCD measures the similarity of the states that belong only to runs of the same system. Because of computational constraints, we perform the analysis with 20 randomly selected runs from the total of 50 runs. Using the notation from the MDS section, we have that r_i denotes the i^{th} run, $\forall i = 1 : 20$ and t_k denotes the time point,

$\forall k = 0 : 29$. We compute the NCD between the states at two time points t_k and t_l, from two runs, r_i and r_j, $\forall k, l = 0 : 29$ and $\forall i, j = 1 : 20$. The state at time point t_k belongs to r_i and the state at time point t_l belongs to r_j.

- The NCD values, $\mathrm{NCD}(\mathrm{r}_i(t_k), \mathrm{r}_j(t_l))$ are plotted on the $x-$axis;
- the NCD values between the consecutive states, $\mathrm{NCD}(\mathrm{r}_i(t_k + 1), \mathrm{r}_j(t_l + 1))$ are plotted on the $y-$axis;
- Let $\mathrm{p}_{\mathrm{NCD}}$ denote the probability mass function of these NCD points. We display $\mathrm{p}_{\mathrm{NCD}}$ on the $z-$axis.

We estimate $\mathrm{p}_{\mathrm{NCD}}$ with the two-dimensional kernel density estimation method developed in [19]. The grid size for the kernel density estimation is 256×256. The probability mass function is estimated over the range of the NCD values $[0\ \ 1] \times [0\ \ 1]$.

If we display the two-dimensional probability mass function of the five setups in three dimensional figures, we cannot obtain any significant visual differences between them. The reason is the fact that the differences are very small and we need a precise method of fine resolution to present those differences in a visible manner. To enhance the variations between these maps of the convergence of the states of the systems, we display contour levels of these probability mass functions in two-dimensional plots. A contour level is equal to the height at a given horizontal section of the two-dimensional probability mass function estimated with kernel density estimation. In this way, we transform the three-dimensional figure showing $\mathrm{p}_{\mathrm{NCD}}$ into a two-dimensional plot. In this transformation, we keep the x and y axes the same, but we indicate the probability values with an array of colours ranging from dark blue, for low values, to dark red, for high values. So, information is not lost, because we display the same information, but in a more intuitive and more clearly visible manner.

In order to create this transformation, we need to solve the problem of optimally selecting the contours, such that they provide a good enough resolution, to be able to distinguish the systems from one another. We cannot use evenly spaced contour levels, because the probability mass functions have a high peak and the rest of the values are very small. There is a small region with very large probability values and many large regions with very small probability values. If we used evenly spaced contour levels, their value would have to be very small to capture the region with small probability. However, this would lead to having too many contours in the region of high probability, such that they would not be visible in the picture. As a solution, we

have developed an algorithm to select unevenly spaced contour levels. The distance between two consecutive levels is small, in the regions of low probability values, and large, in the regions of high probability values. This algorithm is presented in detail in the next section.

Determining the contour level values

The main steps of the contour level selection algorithm are:

- Let $\mathrm{p_r}$ denote the probability mass function for the random setup, $\mathrm{p_{WT}}$ denote the probability mass function for the wild type, $\mathrm{p_{IL\text{-}10}}$ denote the probability mass function for the IL-10 knock-out setup, $\mathrm{p_{IFN\text{-}\gamma}}$ denote the probability mass function for the IFN-γ knock-out setup and $\mathrm{p_{CTLA\text{-}4}}$ denote the probability mass function for the CTLA-4 knock-out setup.
- Set the minimum contour level to $m_{lim} = 0$.
- Set the maximum contour level to M_{lim} to

$$M_{lim} = p_5 - thr_{max}, \tag{4.3}$$

where

$$p_5 = \min\left[M_{p_r} \quad M_{p_{WT}} \quad M_{p_{IL-10}} \quad M_{p_{IFN-\gamma}} \quad M_{p_{CTLA-4}}\right] \tag{4.4}$$

$$\begin{aligned}
&thr_{max} - \text{ is a fixed constant defined by the user,}\\
&M_{p_r} = \max\left[\max\left[p_r\right]\right],\\
&M_{p_{WT}} = \max\left[\max\left[p_{WT}\right]\right],\\
&M_{p_{IL-10}} = \max\left[\max\left[p_{IL-10}\right]\right],\\
&M_{p_{IFN-\gamma}} = \max\left[\max\left[p_{IFN-\gamma}\right]\right],\\
&M_{p_{CTLA-4}} = \max\left[\max\left[p_{CTLA-4}\right]\right].
\end{aligned} \tag{4.5}$$

- Select a number of $N_L = 201$ equally spaced levels between the limits m_{lim} and M_{lim} and discard the first level which is equal to 0. This results in 200 contour level values, which are stored in the vector $\mathrm{EC_L}$.
- Until this step of the algorithm, the operations have been performed for all the systems together. The following steps are executed for each setup separately:

- Compute the coordinates of the points that represent the contours located at the levels stored in the vector $\mathrm{EC_L}$.

- A number of contour levels between 20 and 30 provide sufficient detail and clarity in the display of the probability mass function.

- We quantify how close two contours are by comparing their area. If the ratio of areas is larger than a given threshold, than the two contours are at enough distance apart, to appear clearly on the figure. By varying the threshold, we obtain different number of contour levels. If the value of the threshold is large, the number of the final contour levels is low, if the value of the threshold is small the number of the final contour levels is high. Let C_1 and C_2 be two contours, with areas denoted as A_1 and A_2, respectively. We compute the ratio of their areas as

$$R_A = \begin{cases} \dfrac{A_1}{A_2} & \text{if } A_1 \geq A_2 \\ \dfrac{A_2}{A_1} & \text{if } A_1 \leq A_2 \end{cases} \tag{4.6}$$

  A_1 represents the area of the latest selected contour and A_2 represents the area of each candidate contour that is tested.

- The evenly spaced contours are stored in the vector $\mathrm{EC_L}$. We will select the final smaller number of unevenly spaced contours from this vector. The vector $\mathrm{UC_L}$ contains unevenly spaced contours levels. The algorithm starts with the lowest level from $\mathrm{EC_L}$ and adds it to the new list of contours, $\mathrm{UC_L}$. This is the last entry into the new list. It represents our currently selected contour, against which we will compare contour levels taken from the original list, $\mathrm{EC_L}$. We compare the next contour from the original list to the last entry of $\mathrm{UC_L}$, using the area ratio described above. If this ratio is above the predefined threshold, then we add the candidate contour to the $\mathrm{UC_L}$. Otherwise, we move to the next contour in the original list, $\mathrm{EC_L}$, and compare it to the last entry of the $\mathrm{EC_L}$. This procedure is repeated until all the contours in the $\mathrm{EC_L}$ have been tested.

- Using this procedure, we create a list of unevenly spaced contours for each setup, $\mathrm{UC_L}$. As a result, there are five separate lists of optimal contour lines for each setup. In order to compare the five setups, they must have the same contour levels. We merge all the five vectors that we have obtained for each

setup separately, into one list, denoted as $\mathrm{TUC_L}$. We eliminate the duplicate values. The merged list, $\mathrm{TUC_L}$, contains all the candidate contour levels. From this list, we will select a smaller number of optimally placed levels, such that our initial objective is met.

- The following steps are performed for the combined list of contour levels from all the five setups, $\mathrm{TUC_L}$:

- As the area of a contour at a given level is different for different setups, it is not possible to compare two contours using the ratio of areas as described above. We have to compare the actual values of the contour levels to narrow the list of possible contours. Let $\mathrm{FC_L}$ be the list that contains the final contour levels, which are the same for all the systems. At the start of this procedure, $\mathrm{FC_L}$ contains the first contour level of the merged list, $\mathrm{TUC_L}$. We test each of the candidate contours from $\mathrm{TUC_L}$ against the last entry of the $\mathrm{FC_L}$. The criterion we use for accepting or rejecting a candidate contour level is its relative difference to the current contour, which is the last entry in the $\mathrm{FC_L}$. The candidate contour is accepted if this relative difference is above a given threshold. Let C_1 and C_2 be two contours and their associated levels be L_1 and L_2, respectively. We compute the relative difference between their values as

$$R_C = \frac{|L_1 - L_2|}{L_2}, \tag{4.7}$$

  where $|\cdot|$ denotes the absolute value. L_1 represents the level associated with the candidate contour and L_2 represents the level associated with the latest selected contour.

The experimental parameter values that we used to create the figure 4 of [95]:

- The setups are: the random system, the wild type (WT), the IL-10 knock-out system, the IFN-γ knock-out system and the CTLA-4 knock-out system.

- The number of runs for each setup is equal to $N_{runs} = 20$.

- The number of time steps is equal to $N_{steps} = 30$.

- For a given setup, the NCD is computed between any combination of runs (r_i, r_j), $\forall r_i = 1:20, r_j = 1:20$, taken at two distinct time points, t_k from r_i and t_l from r_j, $\forall t_k, t_l = 0:29$, $t_1 \neq t_2$;

- The grid of NCD values for the probability mass function estimation is $[0\ \ 1] \times [0\ \ 1]$.
- The size of the kernel density estimation grid is 256.

All the parameter values used to create the figure 4 of [95] are determined experimentally, such that the number of final contour levels is between 20 and 30:

- The maximum contour level, M_{lim}, is taken smaller than the minimum of the maximum probability values of the 5 setups, p_5, by the amount equal to $thr_{max} = 0.01$.
- The number of linearly spaced contour levels is equal to $N_L = 201$.
- The threshold used in the first contour selection using the ratio of areas is equal to $thr_{area} = 1.3$,
- The threshold used in the second contour selection using the relative difference between contour levels is equal to $thr_{diff} = 0.15$.

The mapping of probability values to colours

The colour palette used to denote the space between two consecutive contours starts form dark blue to dark red. The blue colours are mapped to small values of the contour levels and the red colours are mapped to large values of the contour levels. The transition of colours from dark blue to dark red is done in ascending order of the contour level values.

The mapping of probability values to colours is done in the following manner: for example, if we have 10 contour levels, the area corresponding to the (x, y) points that have probability values between the first contour level and the second contour level is coloured using dark blue (the first colour in the colour palette). The area corresponding to the (x, y) points that have probability values between the second contour level and the third contour level is coloured using blue (the second colour in the colour palette) and so on, until dark red (the last colour in the colour palette) is used to indicate the area of the (x, y) points that have the probability mass greater than the tenth contour level. The probability mass between two consecutive levels is shown with one colour, which is found on the colour bar at the contour level of lower value in figure 4 of [95].

The results of the probabilistic analysis

We created the probabilistic maps of convergence for the biological conditions 1 and 2, described in the section 4.4. The results for the experimental condition 1 are shown in the figure 4.5. The figure 4.5 was published as the figure 4 of [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`.

We found that the random model shares the least amount of similarities with the wild type and with the rest of the perturbed systems. This is to be expected, as we introduced the random model as a null hypothesis to validate our methodology. The random model does not have any biological significance. Thus, our methodology is validated, as we obtain a clearly distinct shape of the two-dimensional map of the behaviour of the random model compared to those of the rest of the systems. The maps of the convergence of the states illustrates that the CTLA-4 knock-out system is the closest to the wild type, followed by the IL-10 knock-out system and, lastly, by the IFN-γ knock-out system. This system exhibits the most dissimilar dynamics, out of all the perturbations, when compared to the wild type. We obtained the same conclusions using the MDS analysis.

The results for the experimental condition 2 are shown in are shown in the figure 4.6. The figure 4.6 was published as the supplemental figure 3 of [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`,
`http://journals.plos.org/plosone/s/content-license`.
Here, the probabilistic analysis also supports the conclusions of the MDS analysis. As the knock-out efficiency is raised from 25% to 100%, in linear increments of 25%, the similarity of the shape of the convergence map for the IL-10 partial knock-out system decreases compared to that of the wild type. Thus, the system dynamics become more different, as the IL-10 cytokine becomes totally removed from the system.

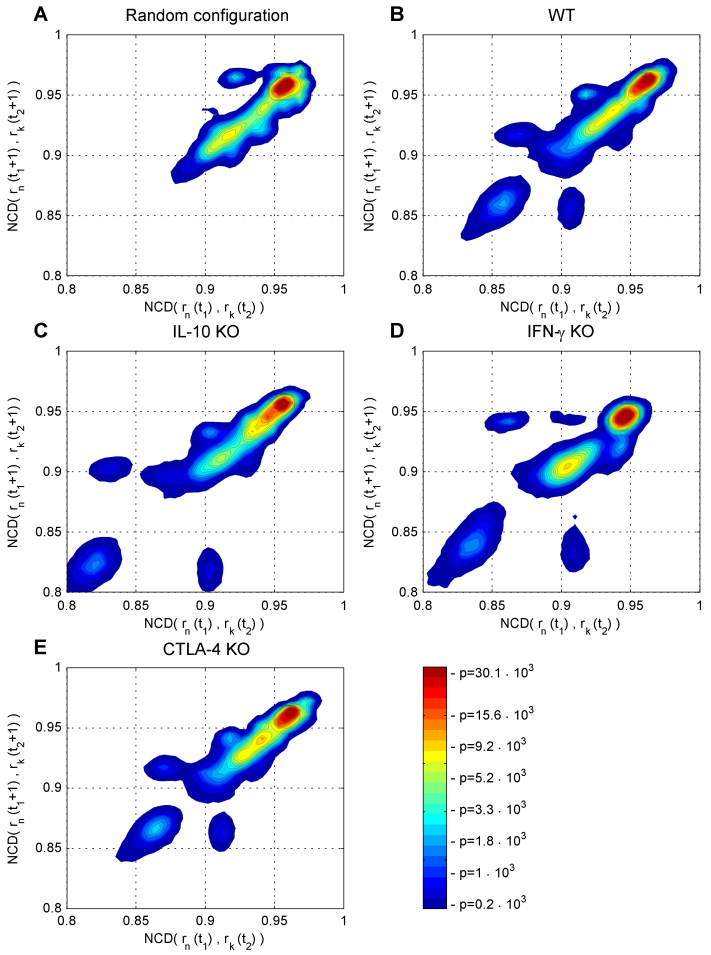

Figure 4.5: This figure was published as the figure 4 in [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`. The probabilitstic maps of convergence for the random setup, the wild type, the IL-10, IFN-γ and CTLA-4, all the perturbations at 100% efficiency.

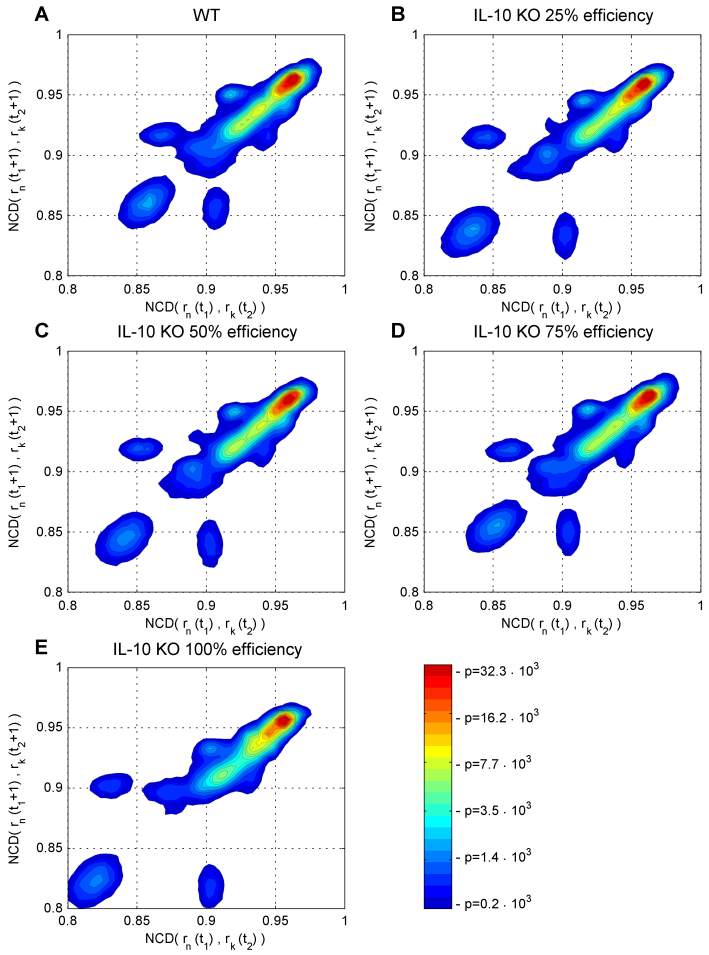

Figure 4.6: This figure was published as the supplemental figure 3 in [95], under the Creative Commons Attribution (CC BY) license `http://creativecommons.org/licenses/by/4.0/legalcode`, `http://journals.plos.org/plosone/s/content-license`. The probabilistic maps of convergence for the wild type and several knock-out perturbations of the IL-10, at different efficiencies.

Chapter 5

The structure and the dynamics of random Boolean networks

5.1 Models of gene regulatory networks

Gene regulatory networks are prime examples of complex systems. As such, their analysis and synthesis presents many challenges. Creating the model of a genetic regulatory network involves an iterative process of simulating the model, making predictions based on the results of the simulations, testing these predictions on the actual gene regulatory networks, through biological experiments, comparing the results of the predictions of the simulations with the results of the biological experiments, changing the parameters of the models to accommodate the difference between the predictions and the experimental results [28]. Gene regulatory networks can be described by several model classes. Some of the criteria used to select the most appropriate model class are: the level of detail of the representation, the amount of available experimental data and computational power and whether the aim is to answer a qualitative question about the overall, global behaviour of the system or a specific, quantitative one about some parameters of the system [53]. Discrete and continuous models of genetic regulatory networks include: directed and undirected graphs, Bayesian networks, Boolean networks, generalized logical networks, nonlinear ordinary differential equations, piecewise-linear differential equations, qualitative differential equations, partial differential equations, stochastic master equations and rule-based formalisms [28]. The different categories of models are characterized by a combination of the some of the following features: static or dynamic, discrete or

continuous, deterministic or stochastic, qualitative or quantitative, coarse-grained, average-grained or fine-grained.

Random Boolean networks were introduced by Kauffman [57] to model gene regulatory networks, in an attempt to uncover the mathematical laws that govern gene regulation. They are discrete, deterministic, qualitative and coarse-grained models of gene regulation. Although a simple model, many of its properties are not fully understood yet, so it is still the subject of intensive research. Random Boolean networks, as well as several other models of complex systems, such as maps that give rise to chaos and cellular automata, require simple equations and a few number of rules to describe them, but, have unexpected and complicated dynamical behaviour. Their dynamics are not trivial to understand and a great deal of research effort is needed to explain how they function and what is the relationship between their structure and their dynamics. Additional difficulties in their analysis include the fact that the state space cannot be explored exhaustively for networks with a large number of nodes, typically above 100. Thus, computational studies need to be carefully implemented, in order to obtain meaningful conclusions and predictions. Moreover, analytical results are very difficult to derive, in general, for any type of Boolean network. Describe here what advances have been made with information theory, how information theory has managed to solve some of these problems.

5.2 The structure of random Boolean networks

In this study, we investigate synchronous directed random Boolean networks. That is, the states of all the nodes of the network change at the same time and if a connection is present from one node to another, it may not exist in the opposite direction. There are also other types of Boolean networks, of which we mention two important variations: the asynchronous Boolean network model [49], which has an asynchronous updating scheme of the state of the nodes, and probabilistic random Boolean networks, which are stochastic models [103].

Throughout this thesis, we will refer to the synchronous directed random Boolean network model, as the random Boolean network model. We present the definition and properties of the structure and the dynamics of random Boolean networks, using the mathematical notation of [100]. A random Boolean network is composed of N nodes, denoted as O_i, $\forall i = 1 : N$. Each node has a number of incoming edges from other nodes in the network and a number of outgoing edges to other nodes in the

network. The number of incoming edges is termed *the in-degree of a node* and the number of outgoing edges is termed *the out-degree of a node*. These quantities can be either fixed, i.e. the same for each node of the network, or variable, i.e. different for each node of the network, but coming from a specified distribution. For example, we investigate networks with in-degrees and out-degrees coming from Poisson and scale-free distributions. The case where the in-degree or the out-degree is fixed can be considered as a distribution with probability equal to 1 that the node will have the fixed in-degree or out-degree. The in-degrees and the out-degrees of the nodes of the network have the following property: the mean value of the in-degree has to be equal to the mean value of the out-degree. We use this property when we generate the networks in the following manner: we select a value for the mean in-degree. This value will also be given to the mean out-degree. This is necessary to ensure that all the in-coming and out-going links in the network are connected. To make the explanation more clear: let N_i be the total number in the network of in-coming edges, N_o be the total number in the network of out-going edges, $\overline{K_{in}} = \overline{N_i}$ the mean in-degree and $\overline{K_{out}} = \overline{N_o}$ the mean out-degree. Then,

$$N_i = \sum_{i=1}^{N} N_{i_{O_i}}, N_o = \sum_{i=1}^{N} N_{o_{O_i}} \text{ and } \overline{N_i} = \frac{N_i}{N}, \overline{N_o} = \frac{N_o}{N}. \tag{5.1}$$

For the network structure to exist, the condition $N_i = N_o$ must hold, or, alternatively, $\overline{N_i} = \overline{N_o}$.

Generating algorithm for the structure of the network

Since all the distributions investigated in this study are discrete, we will refer to the distribution functions as probability mass functions in the remainder of this description. The Poisson probability distribution of a discrete random variable X, with ensemble $\mathcal{E}_X$, is of the form [51]:

$$p_X(x) = \frac{\lambda^x}{x!} \cdot e^{-\lambda}, x \in \mathcal{E}_X = \{0, 1, 2, \ldots\}, \tag{5.2}$$

λ is a parameter, with $\lambda > 0$. The mean and the variance of the Poisson distribution are both equal to λ, i.e. $\mathbb{E}X = \lambda$ and $\mathrm{Var}(X) = \lambda$, where $\mathbb{E}$ denotes the expectation operator.

The scale-free network model, with its characteristic power-law degree distribution [80], [11], [17] is a better model than the Erdős-Rényi random graph model

[34],[35] to describe real-world complex networks. In addition to these models, graph models can also have arbitrary degree distributions [81]. In this study, we use arbitrary degree distributions, as we specify the in-degree and the out-degree distributions and, then, we generate the connections between the nodes to create the network.

The power-law distribution for a discrete random variable, X, with ensemble $\mathcal{E}_X = \{1, 2, \ldots, N\}$ is given by:

$$p_X(x) = C \cdot x^{-\gamma}, \tag{5.3}$$

where γ is a parameter, N is the number of nodes in the networks and C a constant, such that that probability mass function sums to 1, i.e.

$$\sum_{x=1}^{N} p_X(x) = \sum_{x=1}^{N} C \cdot x^{-\gamma} = 1 \Rightarrow C = \frac{1}{\sum_{x=1}^{N} x^{-\gamma}}. \tag{5.4}$$

As there are N nodes in the networks, we truncate the probability mass function up to N.

This generating algorithm is not particularly designed for Boolean networks, but it can be applied to any directed network, to create a structure with prescribed in-degree and out-degree distributions. In order to create such a network, we need to assign an in-degree and an out-degree to each node of the network and, then, connect the links between the nodes. We select the type of probability distribution for the in-degree and that for the out-degree. For each node, we draw two numbers: one from the in-degree probability mass function and one from the out-degree and assign them to each node. Then, we randomly connect all nodes in the network.

The design of the structural classes

We investigated the following structural classes of Boolean networks, having different combinations of in-degree and out-degree distributions and of types of Boolean functions:

1. In the first set of ensemble of Boolean networks, for each node, the function is drawn at random from the uniform distribution on all Boolean functions with a given number of inputs:

- fixed $K_{in} = 3$ in-degree of each node and fixed $K_{out} = 3$ out-degree of each node;
- Poisson in-degree distribution, with $\overline{K_{in}} = 3$, and Poisson out-degree distribution, with $\overline{K_{out}}$;
- scale-free in-degree distribution, with $\overline{K_{in}} = 3$, and scale-free out-degree distribution, with $\overline{K_{out}}$;
- fixed $K_{in} = 2$ in-degree of each node and fixed $K_{out} = 2$ out-degree of each node;
- modular network composed of 3 modules: the first module having fixed $K_{in1} = 3$ and fixed $K_{out1} = 3$, the second module having fixed $K_{in2} = 4$ and fixed $K_{out2} = 4$ and the third module having fixed $K_{in3} = 5$ and fixed $K_{out3} = 5$.

2. In the second set of ensembles of Boolean networks, the topology is a fixed $K_{in} = 3$ in-degree and a fixed $K_{out} = 3$ out-degree, for each node:

   - canalizing Boolean functions [104] drawn at random from the uniform distribution on all canalizing Boolean functions and noncanalizing Boolean functions drawn at random from the uniform distribution on all noncanalizing Boolean functions;
   - canalizing Boolean functions drawn at random from the uniform distribution on all canalizing Boolean functions and Boolean functions drawn at random from the uniform distribution on all Boolean functions.

The degree generating algorithm:

1. We select the type of in-degree distribution and we select its mean.
2. We select the type of out-degree distribution, with its mean equal to the in-degree mean.
3. For each node $O_i, i = 1 : N$, we draw two numbers: one from the in-degree distribution and one from the out-degree distribution (topic: how to draw numbers from an arbitrary probability distribution explain!). We assign the first number as the in-degree of the node and the second number as the out-degree of the node.

4. As the network is finite, we obtain that $\sum_{i=1}^{N} N_{i_{O_i}} \neq \sum_{i=1}^{N} N_{o_{O_i}}$. To solve this problem, we execute the following steps:

5. We repeat step 3 for $N_{trials} = 100$ times and we select the network that has the smallest gap between the sum of the in-degrees and the sum of the out-degrees: select the j^{th} network, with the property that

$$\min_{j} \left| \sum_{i=1}^{N} N_{i_{O_i}} - \sum_{i=1}^{N} N_{o_{O_i}} \right|, \forall j = 1, 2, \ldots, N_{trials}, \tag{5.5}$$

where $|\cdot|$ denotes the absolute value.

6. If $\sum_{i=1}^{N} N_{i_{O_i}} > \sum_{i=1}^{N} N_{o_{O_i}}$, then we need to select several nodes of the network to change their out-degree with a larger value than they currently have, to reduce that gap between the sum of the in-degrees and the sum of the out-degrees. In the end, the gap will be 0. This is the aim of this section of the algorithm. We randomly select a node, from the uniform distribution on all the nodes of the network, to change its out-degree. We keep drawing numbers from the out-degree distribution, while we get a number that is smaller than the current out-degree for the selected node or while replacing the old out-degree with the new out-degree yields a sum of the out-degrees that is larger than the sum of the in-degrees. Once we have found the proper new out-degree, we change the old out-degree of the currently selected node with the new out-degree and we move on to randomly select another node from the network to change its out-degree. It can be the same node, as we select them at random from the uniform distribution on all the nodes of the network, as long as the above mentioned conditions are met. We repeat the procedure described in this step until $\sum_{i=1}^{N} N_{i_{O_i}} = \sum_{i=1}^{N} N_{o_{O_i}}$.

7. If $\sum_{i=1}^{N} N_{i_{O_i}} < \sum_{i=1}^{N} N_{o_{O_i}}$, then we need to select several nodes of the network to change their in-degree with a larger value than they currently have, to reduce that gap between the sum of the in-degrees and the sum of the out-degrees. In the end, the gap will be 0. This is the aim of this section of the algorithm. We

randomly select a node, from the uniform distribution on all the nodes of the network, to change its in-degree. We keep drawing numbers from the in-degree distribution, while we get a number that is smaller than the current in-degree for the selected node or while replacing the old in-degree with the new in-degree yields a sum of the in-degrees that is larger than the sum of the out-degrees. Once we have found the proper new in-degree, we change the old in-degree of the currently selected node with the new in-degree and we move on to randomly select another node from the network to change its in-degree. It can be the same node, as we select them at random from the uniform distribution on all the nodes of the network, as long as the above mentioned conditions are met. We repeat the procedure described in this step until $\sum_{i=1}^{N} N_{i_{O_i}} = \sum_{i=1}^{N} N_{o_{O_i}}$.

8. We create the connectivity matrix of the network, denoted as CM. This step represents the wiring of the network, after all the nodes of the network have an in-degree and an out-degree assigned to them, which come from their specified distributions and which satisfy the condition that $\sum_{i=1}^{N} N_{i_{O_i}} = \sum_{i=1}^{N} N_{o_{O_i}}$. In the beginning of the wiring algorithm, each node needs a number of incoming links to it from other nodes and a number of outgoing links from it to other nodes. We go through all the nodes of the network, $O_i, \forall i = 1 : N$. Whenever we move ahead to a new node, all its incoming and outgoing links are not connected to other nodes. For each node, we randomly select nodes from the uniform distribution on all the nodes of the network and connect them to the current node, such that the connections satisfy the number of incoming and outgoing links that the current node needs. If the selected nodes do not have any unconnected incoming or outgoing links, we discard them and we randomly select other nodes from the network that still have some remaining free connections. There will always be such nodes, because the condition $\sum_{i=1}^{N} N_{i_{O_i}} = \sum_{i=1}^{N} N_{o_{O_i}}$ holds for the entire network. The wiring algorithm ends when all the nodes have been connected and there are no more free links.

5.3 The dynamics of random Boolean networks

The random Boolean network model has the structure described in the previous section. Its dynamical behaviour is determined by the state of each node of the network, which can be either 0 or 1, and by the Boolean function assigned to each node. *A trajectory of a node* represents a vector of states, with elements 0 and 1, from time step $t = 0$ up to the time step when the simulation is stopped, $t = N_s$. *A trajectory of the Boolean network* represents a matrix of 0 and 1, formed by the concatenation of the trajectories of all the nodes of the network. It will have a size of $N_s \times N$, where N_s is the number of time points of the simulation and N is the number of nodes of the network. *A Boolean function* of n variables, $f : \{0,1\}^n \rightarrow \{0,1\}$, takes values from the set of all the combinations of 0 and 1 of size n and returns the value 0 or 1. The number of Boolean functions with n variables is equal to 2^{2^n}. Each node of the Boolean network, has a Boolean function associated with it.

The network starts in an initial state, where all the nodes have the state either 0 or 1. At each time point, all the nodes change their state according to their Boolean function and to their input nodes, as specified by the connectivity matrix of the network, denoted as CM. To illustrate this fact, we take one node of the network, O_i, as an example. Let $N_{i_{O_i}}$ be the neighbours of the node O_i that have out-going edges to O_i. These are the in-coming links to O_i and whose number is equal to the in-degree of O_i. The state of the input nodes to node O_i, at a generic time point t, is contained in the vector:

$$\left[O_i^1(t) \dots O_i^{N_{i_{O_i}}}(t) \right]. \tag{5.6}$$

Let $O_i(t+1)$ denote the state of the node O_i at the next time point, $t+1$. Then, we have the following update rule:

$$O_i(t+1) = f\left(\left[O_i^1(t) \dots O_i^{N_{i_{O_i}}}(t) \right] \right). \tag{5.7}$$

Let p be the probability of a Boolean function to have a 0 in its output table. The output table refers to the values of the Boolean function, given the values of its inputs. For a certain combination of the inputs, the Boolean function can be either 0 or 1. The output table consists of all the particular values the Boolean function can take, for each of its possible input combinations of 0 and 1. If $p = 0.5$, then the function is called unbiased, because the values 0 and 1 have the same probability to appear in the coin flip. The unbiased case is identical to the case when the functions

are drawn from the uniform distribution on all Boolean functions with K inputs. When $p = 0.5$, the network is called an *unbiased network* and when $p \neq 0.5$, the network is called a *biased network*.

The dynamical regimes of the random Boolean networks are termed *ordered*, *critical* and *chaotic*. The change of the dynamical behaviour of a system from one regime to another is named a *phase transition*. The critical regime is the boundary region between the ordered regime and the chaotic regime, where a phase transition takes place between these two regimes [106]. These areas are defined in terms of how perturbations propagate in the network: in the ordered regime, perturbations in the initial state of the network disappear after a few time steps and a large portion of the nodes become frozen, which means that they do not change their state from that point onwards. In the chaotic regime, small perturbations in the initial state of the network propagate and become larger after a few time steps. This is a dynamical phase where the network is extremely sensitive to small changes in its initial state. If the initial state of the network is slightly perturbed, the long-term dynamical behaviour of the network cannot be predicted from this initial state. The critical regime lies at the boundary between the ordered and the chaotic phases. It has the property that the size of the perturbation in the beginning of the experiment remains approximately the same throughout the time series. The perturbation does not die out or become amplified, as the network is run forward in time.

The random Boolean networks introduced by Kauffman [57] are undirected networks, with a fixed in-degree K and the function bias equal to $p = 0.5$. The authors of [30] analytically derived the value of the critical connectivity $K = K_c$, which establishes when the phase transition takes place in these networks:

$$K_c = \frac{1}{2 \cdot p \cdot (1 - p)}. \tag{5.8}$$

The same properties have been derived for generalized Kauffman networks [106]. These are random Boolean networks with the input degree for each node coming from a probability distribution with the mean $\overline{K}$ [106], instead of being fixed, as in the traditional Kauffman networks [57]. In this case, the critical mean connectivity $\overline{K_c}$ that shows when the phase transition takes place is equal to

$$\overline{K_c} = \frac{1}{2 \cdot p \cdot (1 - p)}. \tag{5.9}$$

These properties are generalized as an order parameter of the network, named

the expectation of the average sensitivity [104]

$$s = 2 \cdot K \cdot p \cdot (1 - p) \,. \tag{5.10}$$

The structure to dynamics relationship is quantified by this order parameter s. Given the structural parameters K and p, which characterize the structural class of the random Boolean network, this equation indicates the dynamical regime, or the class of dynamics, of the network. In addition, it can be interpreted as the average number of nodes to which a perturbation can propagate in a network. For example, if $s = 1$, a perturbation in the network propagates, on average, to only one node, thus the perturbation never dies out nor does it get amplified. This value is specific to critical networks. If $s < 1$, then perturbations propagate to less than one node, on average, which means they disappear in the long term dynamical behaviour of the system. These characteristics belong to ordered networks. If $s > 1$, then perturbations become amplified in the network, because they spread to more than one node, on average, which characterizes chaotic networks. The Lyapunov exponent λ is another order parameter of random Boolean networks [71], closely related to the expected average sensitivity [104], by the equation $\lambda = \log(s)$.

Attractors

The properties of the dynamical states of Boolean networks can be described in terms of their *attractors*. A Boolean network of N nodes has a total of 2^N states, because each node can be in a state that is either 0 or 1. The enumeration of all these states represents *the state space* of the Boolean network [2]. The state space of a finite Boolean network is finite. As a result, when run forward in time, the Boolean network will pass through the same state multiple times [2]. A sequence of states that are repeated is termed *an attractor*. The states of the network that lead to an attractor constitute *the basin of attraction*. A network can have multiple attractors. The properties of the attractors of scale-free networks are discussed in [2] and those of the networks with fixed N and K (the standard $N - K$ Kauffman models [57]) can be found in [3].

The dynamical behaviour of random Boolean networks is traditionally studied by measuring the number and length of its attractors. The attractor length represents the number of states that constitutes the attractor. As the random Boolean network is a random model, the approach is to analyse ensembles of such networks and derive

the mean value (the expectation or the expected value) of the properties under study. Two dynamical quantities are of interest: *the mean length of the attractor* and *the mean number of the attractors* of a particular model of random Boolean networks.

The properties of the attractors have been analysed both numerically and analytically. Numerically, the dynamical behaviour suffers from the problem of undersampling. This refers to the fact that computer simulations cannot sample all states or relevant states, because the state space is very large. It is not possible to know if some or all of the states of an attractor have been sampled or not. As a result, the entire state space of a Boolean network cannot be investigated exhaustively. Thus, analytical studies of random Boolean network become extremely important in the understanding of the networks' dynamical behaviour and in designing better simulation experiments. However, analytical results are very difficult to obtain, despite the mathematical simplicity of the Boolean network model.

Numerical studies of random Boolean networks started with Kauffman, when the model was introduced in [57]. But, it was not until after the year 2000 that analytical result were proven. Kauffman numerically found that the mean number of attractors or cycles in a network is a function of the square root of the size of the network, N, [57]. The estimate of the mean number of attractors is improved in [15] to a linear function of N, for critical Boolean networks with a fixed number of inputs equal to 2,3 and 4. However, these numerical results are proven incorrect, by analytical means in [97]. There, the authors analytically study the standard $N-K$ Kauffman model [57], with $K=2$ and equal probability for the Boolean update functions, thus, having critical networks. They define the attractors as cycles of states of length L, termed *L-cycles*. Using an ensemble approach, they prove that the mean of the length of the L-cycles is lower-bounded by a power-law in N, which means that the mean attractor length grows faster than this power-law, as N tends to infinity. They perform computer simulations and show that these simulations produce a biased estimate of the mean attractor length. They obtain the $\sqrt{N}$ result that was previously reported in the literature [57]. This result is incorrect, as the authors prove analytically that the exponent of N in the power-law function is different from $\frac{1}{2}$, which is the exponent of the $\sqrt{N}$. For more information on different numerical results on the mean attractor length, we refer the reader to the references in the articles [97] and [32].

Several other analytical studies of random Boolean networks have been conducted by various authors. The authors of [33] investigate the mean number of attractors

and the mean lengths of attractors in the case of critical random Boolean networks with N nodes and $K = 1$. They analytically derive a lower bound for the mean attractor length, which is a power-law function of N and they give an approximation of the mean number of attractors. The interesting result is that the mean attractor length grows faster than a power-law function of N, which is a similar result obtained by [97], for critical random Boolean networks with N nodes and $K = 2$. The author of [32] applies the analytical techniques of [97] to show that the mean number of attractors can be approximated by a power-law function of N, in the case of critical random Boolean networks with $K = 1$. The authors of [58] derive lower bounds for the mean number of cycles in different types of modular Boolean networks with loops and show that the mean number of cycles grows faster than a power-law function of N, as N tends to infinity. All these numerical and analytical difficulties motivate the continuous research effort that is carried out in this field of random Boolean networks. A new approach of studying the dynamics of Boolean networks is with information-theoretic methods.

5.4 Previous information-theoretic studies of structure-dynamics relationships in random Boolean networks

The work presented in this thesis continues and improves on several information-theoretic studies of random Boolean networks, which will be described at length, in this section. By quantifying the structure-dynamics relationships with information-theoretic equations different types of ensembles of random Boolean networks are classified into ordered, critical and chaotic networks [86]. The normalized compression distance (NCD) [68], [22] from algorithmic information theory [60], represents the similarity measure used to separate the networks. The NCD is computed between the structure of pairs of networks from the same ensemble and shown in a two-dimensional figure against the NCD between their corresponding dynamics. The critical ensemble with random connectivity displays the most diverse dynamics, compared with ordered or chaotic networks with random wiring and the same networks, having the same degree, but with regular wiring instead of random. This article gives evidence that the highest amount of complexity can be observed in the networks at the critical phase, that is, at the phase transition between ordered and chaotic.

The previously mentioned study is extended to real biological systems in [85].

There, the newly introduced NCD based Derrida curve indicates that the macrophage operates in the critical regime. This plot is based on the NCD measure and it represents a version of the Derrida curve derived for random Boolean networks, to determine their dynamical regime, given some structural parameters. Traditionally, the dynamical regime of such networks is investigated by means of the Derrida curve [30], [31], constructed as a result of a perturbation analysis. The state of the network is perturbed by small amounts and the size of the perturbation is measured at the next time point in the dynamical evolution of the network. The Derrida curve is constructed by plotting the size of the current perturbation against that at the next time point. The slope of the Derrida curve at the origin indicates the dynamical regime of the network. If the slope is smaller than 1, then the network is ordered. If the slope is equal to 1, then the network is critical. If the slope is larger than 1, then the network is chaotic. The NCD extension of this curve is constructed with the NCD applied to experimental data that measure the dynamical states of the macrophage. The state of the gene network of the macrophage is measured with microarray experiments of gene activity, for several time points. The NCD is computed between the network states, under several experimental conditions, at two different time points. Then, the NCD is computed between the same pair of states, but taken at the next time point and displayed against the previous NCD value. Continuing for all time points for which experimental data are available, the authors create an NCD based Derrida curve. It displays how the difference between the states of the network evolves in time. It has the advantage that it can be applied directly to experimental data, instead of a traditional perturbation analysis required for the original Derrida curve. This perturbation analysis is not feasible for real biological systems, for which the investigation of the gene activity is limited to a small number of time points. The authors validate this new method as an extended Derrida curve, by applying it to random Boolean networks.

Based on the normalized information distance (NID) [13], [68] and on its computable version, the NCD [68], [22], *set complexity* is defined as a context-dependent information-theoretic measure of the complexity of strings [43]. It evaluates the cumulative complexity of strings in a set, by taking into account the complexity of the individual strings of the collection, as well as that of the relationship between the strings. Its primary application is to quantify biological information. As such, both identical strings and totally random ones bring no contribution to the overall complexity of the set, meaning that their contribution is zero in the sum of complexities.

Here and in the context of biological systems and biological information, objects that share nontrivial similarities are designated as complex. In contrast, identical strings bring no novel information to the set and random strings have no biological function. The motivation behind the definition of set complexity using Kolmogorov complexity is that it is more adequate than Shannon's information theory [102], [25], to define such a measure of complexity. In the biological context, empirical probability distributions are extremely difficult to construct, because of the large size of the systems and the few available measurements of their dynamical behaviour. In such cases, computing the complexity of individual strings is more straightforward, than modeling them as random sources and deriving probability distributions to characterize them. As an application, biased random Boolean networks, ranging from ordered, critical to chaotic, are shown to have the highest average set complexity of their dynamical behaviour in the critical regime.

For ordered and critical random Boolean networks, the complexity of their trajectories, measured by the set complexity [43], is maximized in the region before the RBN settles into an attractor [72], [73]. But, for chaotic networks, no extreme points of the complexity as a function of time can be distinguished. As there are no short-term or long-term patterns, the dynamical behaviour of chaotic networks resembles random noise. As a result, the set complexity does not exhibit any fluctuations. Random Boolean networks with fixed in-degree are investigated in [72] and with a Poisson in-degree distribution in [73]. In addition, noise added to the dynamics increases the complexity of the trajectories [73]. The noise is modelled as a probabilistic change of the state of a node, after it has been updated using the Boolean function assigned to it [87].

The authors of [93] investigate which types of random Boolean networks contain multiple ergodic sets and what dynamical conditions are needed to have such sets. Two structural classes are analysed: synchronous unbiased random Boolean networks with random and scale-free topologies. Noise is modeled as a change of the correct state of a node, after updating, to the opposite state, with a small probability of flipping. An ergodic set is defined as the set of states taken from multiple attractors, such that, when a node is perturbed, the new state of the network belongs to the same set of states, that is, the ergodic set. An ergodic set is essentially a partitioning of the state space, and implicitly of the attractors, into groups, such that, when a network enters into an ergodic set, it cannot leave it, due to random perturbations. In addition, any attractor within an ergodic set can be reached from any other

attractor within the set, by gene perturbations. In the two types of structural classes, the perturbation of all the genes results into one ergodic set. However, when only a small fraction of the genes are perturbed, the networks from the two structural classes contain several ergodic sets. As more genes are perturbed, the number of ergodic sets is greater than 1, but it decreases sharply to 1, on average.

The average pairwise mutual information from Shannon's information theory is maximized in the critical regime for synchronous random Boolean networks, for both biased and unbiased random Boolean networks [94]. The mutual information is time-delayed, meaning that the state of the first node is taken at time t and the state of the second node from the pair is taken at time $t + 1$. This time-delayed pairwise mutual information is computed for all the pairs of nodes of the network and is averaged over the entire network.

The basin entropy [61] is a network parameter that measures the complexity of the dynamics of ensembles of random Boolean networks, based on the distribution of basins of attraction of these ensembles. The basin entropy is maximized for critical networks, where it increases with the size of the system. However, in the case of ordered and chaotic networks, it remains bounded as the network size increases to infinity. The average basin entropy is estimated from simulated time series [62], generated for a Boolean network model of the protein interaction network in the mammalian cell cycle [39].

The authors of [59] employ Fourier analysis for Boolean functions, to show that the mutual information between a Boolean function of a sequence of mutually independent random variables and one of its inputs is maximized for canalizing Boolean functions, which are canalizing in that particular input variable. In this case, the mean of the functions over which the maximization is performed is fixed. The authors extend this mutual information to the multivariate case, where they prove that it is maximized by jointly canalizing Boolean functions. In addition, in the univariate case, the authors show that, if the mean of these Boolean functions is not fixed, the same mutual information is maximized by a function that depends only on one variable. This type of Boolean function is termed a dictatorship function. The mutual information is computed between the output of the dictatorship function and the variable which this function depends on.

5.5 Experimental order parameter of the dynamics of random Boolean networks

To quantify the relationship between the structure and the dynamics of random Boolean networks, we develop an experimental order parameter, based on the NCD applied to the time series of network states. It represents an NCD based generalization of the Derrida curve [30], [31] to dynamical data obtained from simulations of real-world biological experiments. We extend the NCD based Derrida curve applied to random Boolean networks in [85], which was used to validate the conclusions found for the macrophage system. In [85], the NCD is applied to two states of the network at time point t and displayed against the NCD between the same states at the next time point $t+1$. Other figures for a time point difference large than 1, that is, $\Delta t > 1$, are given in the supplemental material of the paper, supporting the same conclusions obtained with $\Delta t = 1$.

There is a need to develop an experimental curve similar to the original Derrida curve, to identify the dynamical regime of the network, when only limited dynamical information is available. Because, in the case of real biological systems, we do not have access to a large number of measurements of the state of the system and we cannot perform a perturbation analysis. Our goal is to find novel ways of sampling the state space and to compute an experimental order parameter to indicate the dynamical regime of the network, with the same role as the slope of the traditional Derrida curve has in the theoretical case, where perturbations of the system's state can be performed without any restrictions. We extend the NCD based Derrida curve applied in [85], by computing the NCD between two states of the same system, taken at two different time points and by plotting this value against the NCD between the same states, taken at the following time points. This approach requires fewer states that the one used in [85] and it does not involve a perturbation analysis. We prove that it can distinguish the dynamical regime for several structural configurations of the random Boolean network model. In addition, we develop an experimental order parameter, based on these curves, which is extremely useful in indicating the dynamical properties of these networks, using only experimental data.

The description of the RBN model

We simulated time series data for the random Boolean networks with a fixed in-degree K, a function bias p and random connections between the nodes. The range of the fixed in-degree K is from 1 to 10. We analysed both unbiased and biased networks, by varying the function bias p in the range of $0.01 - 0.5$, with a step size of 0.025:

- one set of simulation experiments was performed for unbiased networks, that is, $p = 0.5$, by varying the fixed in-degree K;
- the second set was performed for biased networks with the fixed in-degree $K = 3$ and a variable p.

The network starts in a randomly selected state. This is achieved by flipping a fair coin for each of the nodes of the network. The state of the entire network is given by the concatenation of the states of all the nodes. The network is run forward in time and its state changes, as the state of each node changes. The state of each node is modified by the value of its associated Boolean function, according to the value of its input nodes, as described in section 5.3. For each network in our simulations, we selected trajectories of 15 states. After 15 time points, the ordered networks reach an attractor and all the states are identical after this point in time. We decided to exclude these states from the analyses, as they do not bring any further information about the behaviour of the system. Thus, we perform the analyses in the transient phase of the dynamics of the networks, where the greatest amount of change takes place.

The analysis method and the experimental order parameter

We build *an experimental NCD Derrida curve* and compute *an order parameter*, as a property of the curve. For different structural classes, we create such two-dimensional curves and we use them to establish the dynamical regime of the networks. The curves are created by plotting NCD values on the x and y axes of a figure and interpolating these values. Let the state of the network be termed $\mathrm{S_{RBN}}$. On the x axis, we compute the NCD between the state of the network at the time point t_1 and the state of the network at the time point t_2, that is, $\mathrm{NCD}(\mathrm{S_{RBN}}(t_1), \mathrm{S_{RBN}}(t_2))$, $\forall t_1, t_2 \in \{1, 2, \ldots, 14\}$. On the y axis, we compute the NCD between the consecutive states of the network, at the time points $t_1 + 1$ and $t_2 + 1$, that is, $\mathrm{NCD}(\mathrm{S_{RBN}}(t_1 +$

$1), \mathrm{S}_{\mathrm{RBN}}(t_2+1)), \forall t_1, t_2 \in \{1, 2, \ldots, 14\}$. The value corresponding to the coordinates $\mathrm{NCD}(\mathrm{S}_{\mathrm{RBN}}(t_1), \mathrm{S}_{\mathrm{RBN}}(t_2))$ and $\mathrm{NCD}(\mathrm{S}_{\mathrm{RBN}}(t_1+1), \mathrm{S}_{\mathrm{RBN}}(t_2+1))$ is denoted as one point on a two-dimensional figure. We construct this experimental curve by polynomial interpolation, with a moving average filter. The degree of the polynomial is 1 and the interpolation method is local regression with weighted least squares. The area between the diagonal of the plot and the curve represents the experimental order parameter.

The results

In order to test this methodology, we consider the structural class of the random Boolean network known. As a benchmark, we quantify the structure to the dynamics relationship with the theoretical order parameter termed *the expectation of the average sensitivity* of the network. It was derived in [104] and explained in detail in the section 5.3. It indicates the dynamical regime of the network, given its structural class. Then, we compare the experimental results with the theoretical ones. The order parameter is denoted as s and is equal to

$$s = 2 \cdot K \cdot p \cdot (1-p)\,. \tag{5.11}$$

The three dynamical regimes are given by the value of s:

- If $s < 1$, then the network is in the ordered dynamical regime;
- If $s = 1$, then the network is in the critical dynamical regime;
- If $s > 1$, then the network is in the chaotic dynamical regime.

Because of the lack of patterns in the dynamical regime of chaotic random Boolean networks, the NCD measure of the similarity of two states has high values, which are in the upper half of the [0 1] interval. Moreover, these values are similar for any pairs of states. This results in the NCD points forming a tight cloud around the diagonal. Thus, the area between the curve and the diagonal is equal to 0. The area measure is also equal to 0 for critical random Boolean networks, because the points are closer and closer to the diagonal, as the network transitions from the ordered regime to the critical regime. However, the shape of the curve distinguishes the two types of networks: for chaotic networks, the points cluster together in a tight cloud around the diagonal, whereas, for critical networks, they form an elongated curve, which is close to the diagonal, but spans the entire [0 1] interval.

By visual inspection of the figures showing different experimental NCD based Derrida curves, random Boolean networks can be classified in two groups: ordered or critical and chaotic. If the networks belong to the first case, the experimental area measure can be employed to further separate them into structural classes of different degrees of order. Larger values of the area indicate increased order in the dynamics of the networks. Therefore, it represents an experimental indicator of the system's dynamical class, equivalent to the network sensitivity s in theoretical studies. For the second category, the area measure is equal to 0, for networks of all degrees of chaos. Thus, it cannot be used to distinguish between them. However, as the dynamical behaviour of biological networks has been found to be either ordered or critical [105], [7], [86], this limitation does not prevent the accurate analysis of real-world networks. We refer the reader to the article [99], for different visualizations of the experimental NCD curve and the area order parameter, for networks that belong to the structural classes described above.

5.6 Mapping dynamical states to structural classes for random Boolean networks

Motivation

We analyse the dynamics from two structural classes of random Boolean networks. The dynamical data is represented by the trajectories of states of the networks. We construct a methodology that can map dynamical states to structural classes, based on information theory. For this purpose, the structural classes are known in advance, as we generate simulated data for ensembles of random Boolean networks from two chosen structural classes. We then create methods, such that we can correctly separate the two networks into their topological classes, using only dynamical data. Once we have obtained a successful mapping, the structural information will no longer be needed in such analyses, only that of the trajectories of the networks. The objective of our methods is not to investigate topological details of a particular network, but to separate networks into structural classes, based on some general, high-level structural features they share. This is the reason why we study ensembles of networks and not individual ones.

The purpose of this study is to show that structural information is hidden in the dynamical behaviour and that we can find this information without inferring the

topology of the network. This aspect is very important to achieve, as the inference of random Boolean networks is not very accurate and is difficult to compute to the desired level of accuracy. Structural information is not available upon direct inspection of the trajectories of the networks. By direct inspection, we mean comparing by any visual means the matrices of raw dynamical data, containing the elements of 0 and 1, which represent the states of the two networks. In such a case, this dynamical information would appear random. However, for each network, the sequence of states is not random, as the topology and the Boolean functions of the network constrain its possible dynamical behaviour. The patterns in which these 0 and 1 are arranged differs for each structural class of the networks, because they have distinct topologies and Boolean functions. This motivates the usage of information-theoretic methods. By applying information theoretic concepts, we devise a statistical methodology to quantify hidden structural information from dynamical data, by investigating the nonlinear correlations of the dynamical behaviour of the nodes of the networks. We find differences in the patterns of the dynamics between the two networks, which have been introduced due to the differences in the patterns of their structures. By information-theoretic means, we are able to create a mapping of dynamical data to structural classes, with very high accuracy results.

The justification for this study is two-fold: on the one hand, we create a dynamics to structure relationship, in random Boolean networks, for theoretical purposes. On the other hand, it has applications to the inference of real biological systems. For example, for a gene regulatory network, limited time measurements of the state of the network are available, which makes the inference of such networks problematic. Our methodology extracts topological information from the dynamics, which would help guide the choice of more targeted inference methods, once the structural class of the network is known. We are able to find structural properties from the dynamics, without inferring the topology. In addition to the dynamical data, this brings more knowledge to the inference problem of random Boolean networks of gene regulatory networks.

The simulation data

The mathematical notation and symbols used throughout this section is identical to the one in [100]. We create the structure of the random Boolean network and assign the Boolean functions to each node. The information related to the design of the

random Boolean network, that is, the topology and the functions of the network, is not directly used in the classification procedure, but indirectly in the simulation of the dynamical behaviour of the networks. As we analyse ensembles of random Boolean networks, we conduct a statistical analysis of the results of the classification accuracy, using 100 experiments.

For each experiment, we create 1000 samples, where each sample is composed of five features. We train a support vector machine (SVM) [23] classifier on 100 samples and we test it on 200 samples. Both training and testing samples are randomly selected from the available samples. The training samples are selected first from the total of 1000 samples. The testings samples are selected from the remaining ones that have not be used for training the classifier. We repeat this procedure 100 times, to obtain 100 experiments. The results we report are averaged over these 100 experiments. To obtain one sample, we generate the structure of the Boolean network, select the Boolean functions for each node and run the network forward in time. For each sample, the network belongs to the same structural class, that is, the probability distribution of the input nodes, that of the output nodes and the types of Boolean functions are the same, but the connections between any pair of nodes are redrawn at random and the Boolean functions are reassigned for each node.

For each sample, we estimate a time-delayed normalized mutual information matrix. We generate the data to estimate this matrix as follows: the connections and the Boolean functions remain fixed, but the trajectories are recreated from a random initial state for $N_n = 100$ trials. In each trial, the network is restarted from a random initial state and run forward in time for $N_s = 100$ time points. As a result, we obtain $N_n \cdot (N_s + 1) = 10100$ dynamical states for the estimation of the normalized time-delayed mutual information matrix.

One trajectory of the random Boolean networks is obtained as follows: we start the network in a random initial state and we run the network forward in time for N_s number of steps. The initial state of the network is a vector of size N containing the initial states of all the nodes of the network. For each node, the initial state is selected at random to be either 0 or 1, with probability 0.5. At the next time point, $t = 2$, each node changes its state according to the Boolean function associated to it and the previous state of its input nodes. For example, if node O_5 has 3 input nodes, $[O_1, O_{40}, O_{51}]$, and a Boolean function f_5, its state at time point $t = 2$ will be given by:

$$O_5(2) = f_5([O_1(1), O_{40}(1), O_{51}(1)]). \tag{5.12}$$

In general, if node O_i has $N_{i_{O_i}}$ input nodes, $\left[O_i^1 \dots O_i^{N_{i_{O_i}}}\right]$, and the Boolean function $f_i : \{0,1\}^{N_{i_{O_i}}} \to \{0,1\}$, then

$$O_i(t+1) = f_i(\left[O_i^1(t) \dots O_i^{N_{i_{O_i}}}(t)\right]), \forall t = 1, \dots, N_s. \tag{5.13}$$

The classification scheme

For each classification task, we analyse two classes of random Boolean networks. They differ either by their structure or by the type of Boolean functions. In terms of their dynamics, the Boolean networks are synchronous, that is, all the nodes update their state at the same time. Our objective is to measure the trajectories of each of the two networks, extract appropriate classification features from these trajectories and use these features to separate the networks into their correct structural class.

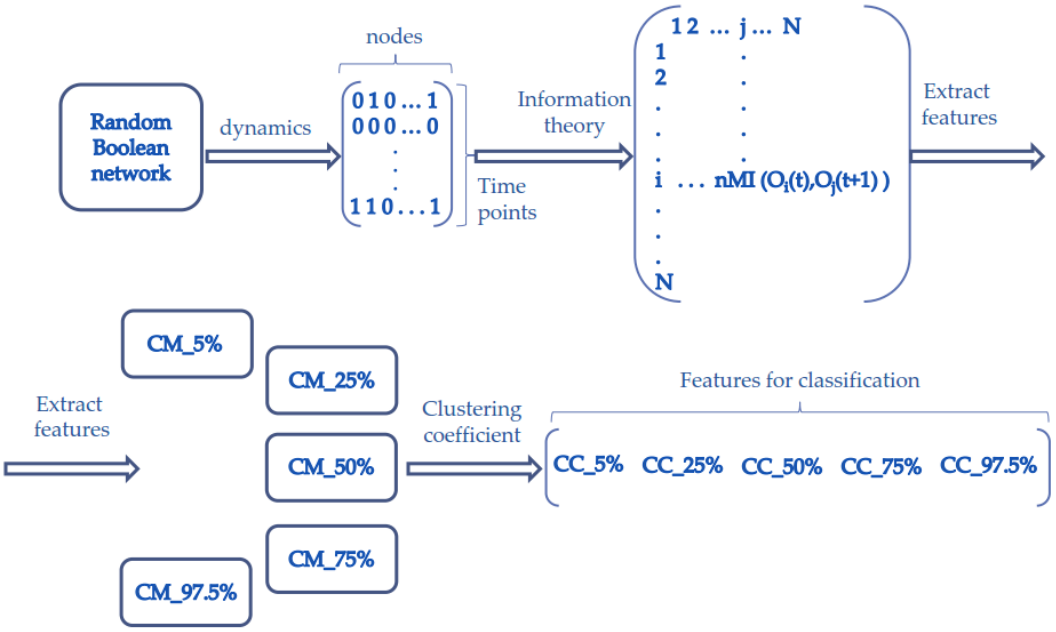

Figure 5.1: The classification procedure of the dynamical states of random Boolean networks

The main steps of our classification procedure are shown in the figure 5.6:

1. We store the dynamical states of the network in an $N_s + 1 \times N$ matrix of 0 and 1.

2. We create a matrix of correlation of the dynamics of the nodes, by computing the time-delayed normalized mutual information between any pair of nodes.

3. We threshold this time-delayed normalized mutual information matrix at five levels, to obtain five approximations of the true connectivity matrix. These matrices contain only values of 0 and 1.

4. We extract one feature for classification from each of these five matrices. A feature is represented by the clustering coefficient of a matrix. We associate a feature vector containing these clustering coefficients, to each trajectory of the network.

5. We train and test a support vector machine classifier (SVM) [23] on the dynamics of two structural classes of random Boolean networks.

In the following paragraphs, we will describe in detail each of the main points of our classification scheme.

Step 1. **Dynamical states**. For each network, we store the trajectories in a matrix, denoted as TS. The size of the matrix is $N_s + 1 \times N$. The rows of this matrix represent the time points, t_i, $\forall i = 1, 2, \ldots, N_s + 1$ and the columns represent the nodes, O_j, $\forall j = 1, 2, \ldots N$. Each element of the matrix, TS(i, j) can be either 0 or 1 and indicates the state of the node O_j, at time point t_i, $O_j(t_i)$. We need to process the information contained in this matrix, to find the most informative features that would enable us to classify the two ensembles of random Boolean networks. The raw information contained in the matrix TS does not provide any means of comparison between networks from the two ensembles. To solve this issue, we use *the time-delayed normalized mutual information* to extract relevant classification features from the TS matrix of dynamical behaviour.

Step 2. **Estimating the time-delayed normalized mutual information matrix**. This mutual information method is a statistical technique of measuring nonlinear correlations between pairs of random variables. We use the same method for the estimation of the time-delayed normalized mutual information matrix, nMI, [100] as that of the authors of [94]. The time-delayed normalized mutual information is computed between any pair of nodes, (O_i, O_j), $\forall i, j = 1, \ldots, N$. The states of the nodes are taken at two different time points, which is why the method is called time-delayed. We use a time delay of one time step, $(O_i(t), O_j(t + 1))$, which is suitable for testing the hypotheses of our study. The time-delay factor is justified

because of the fact that the nodes modify the behaviour of their neighbours from one time point to another, according to their Boolean functions and their current state. Thus, the dynamical behaviour is shaped by the network topology and functions. These elements introduce correlations in the trajectories of the nodes of the network. They are not immediately visible upon direct inspection of the dynamics stored in the matrix TS, but they can be uncovered with information-theoretic methods, as we will prove through this study on random Boolean networks.

One element of the time-delayed normalized mutual information matrix is equal to

$$\text{nMI}(i,j) = \text{nMI}(O_i(t), O_j(t+1)), \forall i,j = 1,2,\dots N. \tag{5.14}$$

We use a plug-in estimation method for each element, $\text{nMI}(i,j)$, $\forall i,j = 1,2,\dots N$. The state of each of the two nodes, $O_i(t)$ and $O_j(t+1)$, is modelled as a discrete random variable that can take values from $\{0,1\}$. To create the time-delayed version of the normalized mutual information between the states of the two nodes, the node O_i is taken at time point t and the node O_j is taken at time point $t+1$. We first estimate their joint probability mass function and, then, we plug-in these values in the formula for the normalized mutual information between the two random variables, as given by the equations 2.29, 2.22, 2.2, from the section 2.1. This section contains the background information on Shannon's information theory. This estimator of the nMI matrix is termed plug-in, because we first estimate intermediate variables, the joint and the marginal probability mass functions, and, then, we compute the final formula with these intermediate values. We compute the joint probability mass function of $(O_i(t), O_j(t+1))$, by counting how many times each pair of states, $(0,0)$, $(0,1)$, $(1,0)$ and $(1,1)$, appears in the joint trajectories of the two nodes. Then, we divide the result by the total number of pair of states. The trajectory of O_i is the collection of states from the time point $t=1$ to $t=N_s-1$ and the trajectory of O_j is the collection of states from the time point $t=2$ to $t=N_s$.

Step 3. **Thresholding the time-delayed normalized mutual information matrix**. We threshold the time-delayed normalized mutual information matrix, using several levels, to obtain matrices that contain only the values 0 and 1. These matrices are approximations of the true connectivity matrix of the network. The connectivity matrix of a network of nodes, or a graph, represents the matrix of 0 and 1, for which the element 1 indicates a direct link between a pair of nodes and 0 shows the absence of such a connection. We denote the connectivity matrix by CM.

In all our experiments, the Boolean networks are directed graphs, as we have input and output nodes for each node and the flow of information from one node may or may not be bidirectional, depending on how the node is connected to other nodes. This means that $\mathrm{CM}(O_i, O_j) \neq \mathrm{CM}(O_j, O_i)$, unless O_i is an input node to O_j and O_j is an input node to O_i. As we do not have access to the true connectivity matrix, we cannot determine the level of accuracy of each of these approximations. Thus, we cannot select one matrix over another, to achieve greater accuracy in our classification scheme. As a result, we combine the information from all these matrices, by computing one feature from each of them.

The thresholding of the time-delayed normalized mutual information matrix produces approximations of the true connectivity matrix of the random Boolean network. If two nodes, O_i and O_j, are connected, then the $\mathrm{nMI}(i, j)$ is expected to have a larger value, than that of other pairs of nodes that are not connected. However, this can be misleading, because the $\mathrm{nMI}(i, j)$ can have a high value, due to reasons other than a direct connection between the two nodes. There can be indirect connections between them, such as the connection of their neighbours, which increases the mutual information between the two nodes. The nodes may be connected, but their $\mathrm{nMI}(i, j)$ may be lower than that of other nodes that are indirectly linked. Values of $\mathrm{nMI}(i, j)$ greater than 0 may appear due to the estimation method, as the estimation error. That is, although the nodes are not connected, we may not obtain an $\mathrm{nMI}(i, j)$ value equal to 0. We cannot expect to obtain perfect results that indicate exactly which nodes are directly connected and which are not. Based on the simulation results we have conducted, we cannot draw a definite conclusion as to the value of the threshold, above which the $\mathrm{nMI}(i, j)$ represents a connection between the nodes O_i and O_j and below which the nodes can be considered not connected.

We select five thresholding levels: the 2.5^{th}, 25^{th}, 50^{th}, 75^{th} and 97.5^{th} percentile of the data from the nMI matrix. The thresholds are computed for each individual nMI matrix. They are the same percentiles of the nMI values, but the actual values of the thresholds are different in each case, as the matrices are different. A k^{th} percentile of a collection of numbers is the value for which $k\%$ of the numbers are below that value. For each percentage, we find the percentile from the numbers of the nMI matrix. This number becomes the thresholding level. All the values of the nMI matrix that are below this level become 0 and the ones that are above this level become 1.

Step 4. **The clustering coefficient as a classification feature**. We select

the directed clustering coefficient as the graph-theoretic measure to describe the matrices obtained as a result of the thresholding scheme detailed at the previous step. As most graph theoretic measures are extremely computationally intensive, we need to select an appropriate one, such that the computational complexity of the classification algorithm is kept feasible, without losing the accuracy. For a description of graph-theoretic measures, we refer the reader to the introductory chapter of this thesis, the section 1.2, where we describe several analysis measures of the structure of complex networks. We chose the directed clustering coefficient because of its fast computation and because it leads to very accurate results. The clustering coefficient was first introduced for undirected and unweighted networks [113], but it can also be implemented for directed and weighted networks [38]. In our study, all the networks are directed networks, so we use the following version of the clustering coefficient:

$$\mathrm{CC_i} = \frac{\sum_{j=1}^{N_{O_i}} \sum_{k=j+1}^{N_{O_i}} [\mathrm{CM}(O_j, O_k) + \mathrm{CM}(O_k, O_j)]}{N_{O_i} \cdot (N_{O_i} - 1)}. \tag{5.15}$$

The average directed clustering coefficient is equal to the mean of the local directed cluster coefficients of the nodes of the network:

$$\mathrm{CC} = \frac{1}{N} \cdot \sum_{j=1}^{N} \mathrm{CC_i}\,. \tag{5.16}$$

As we studied only directed networks, we refer to the directed clustering coefficient simply as the clustering coefficient. The local clustering coefficient is defined as the fraction of connections between the neighbours of a node from all the possible connections between these neighbours. We denote as neighbours both the input and output nodes to the current node. As such, each node O_i has $N_{O_i} = Ni_{O_i} + No_{O_i}$ total number of neighbours. As the link between two nodes is directed, we need to sum both the terms $\mathrm{CM}(O_j, O_k)$ and $\mathrm{CM}(O_k, O_j)$. We need to account for the link between the two neighbours in both directions. If the link exists in both directions, it will account for two links, out of all the possible connections between the neighbours of a node. There are a maximum of $N_{O_i} \cdot (N_{O_i} - 1)$ such connections, because we have N_{O_i} neighbours and, each neighbour can be connected to the other $N_{O_i} - 1$ neighbours.

For each of the five approximations of the connectivity matrix, we compute one average clustering coefficient. Therefore, we have a vector of five elements as five

features for classification. In conclusion, from one dynamical trajectory of a random Boolean network, we create one vector of classification features, which contains five elements.

Step 5. **The classification algorithm**. We classify trajectories of networks from two distinct structural classes, with a support vector machine algorithm (SVM) [23]. We select the following options for the SVM: a Gaussian kernel, a 10-fold cross-validation scheme, a grid search for the optimization of the parameters of the kernel and the missclassification rate as their optimization criterion.

The results

The two types of ensembles of random Boolean networks differ either by their connectivity patterns or by the choice of Boolean functions. The exact details of the design of the structural classes of the Boolean networks are described in the section 5.2. For one classification task, we generate the simulated data as follows: the number of nodes of the network is $N = 100$. For each structural class, we generate $N_e = 1000$ trajectories. We use one subset of these data for training of the SVM classifier and the other subset for testing and reporting the accuracy results.

For the set 1 of types of networks, detailed in the section 5.2, we obtained accuracy results in the range of $94\% - 99.7\%$. For the set 2 of types of networks, detailed in the section 5.2, we obtained accuracy results in the range of $92.8\% - 99.3\%$. For the task of separating the dynamics of Boolean networks that belong to the structural classes of a fixed in-degree $K_{in} = 2$, a fixed out-degree $K_{out} = 2$ and of a fixed in-degree $K_{in} = 3$, a fixed out-degree $K_{out} = 3$, the classification accuracy increases from 93% to 99.5%, as the nodes of the network increase from 50 to 200 nodes. In addition, we conducted a validation test, to see if our method produces correct results in the cases we tested. In this test, we classified trajectories of Boolean networks from the same structural class, in two cases: the first case is represented by networks from the ensemble of a fixed in-degree $K_{in} = 3$, a fixed out-degree $K_{out} = 3$. The second case is represented by networks from the ensemble of a Poisson in-degree distribution with mean in-degree $\overline{K_{in}} = 3$ and of a Poisson out-degree distribution with mean out-degree $\overline{K_{out}} = 3$. The results in both cases are approximately 50% of classification accuracy. This indicates that our classification procedure cannot distinguish between networks that belong to the same ensemble. These results are correct, as we have designed our statistical methodology to separate distinct ensembles of networks, that

is, structural classes, and not individual networks that belong to the same ensemble [100].

Chapter 6

Discussion

The contributions of this thesis are two-fold: on the one hand, we employ methods from information theory, from both Shannnon's information theory and from algorithmic information theory, or Kolmogorov complexity, in novel ways, to find knowledge of the structure-dynamics relationships in an executable biological model of the human immune system, [95], and in the random Boolean network model of gene regulatory networks, [99], [100]. In the first case, the executable modeling of reactive systems shows great promise in describing the structure and the function of complex biological systems. The paradigm of reactive systems and reactive animation has recently been applied to complex biological systems to improve their modeling and visualization. However, suitable analysis methods that can integrate the large amount of data produced by such simulations are lacking. In the second case, random Boolean networks are a well established, tractable model of complex gene regulatory networks. Nevertheless, their properties are not fully understood yet, either by theoretical or computational means and further research is needed in this direction.

On the other hand, we bring theoretical contributions to the generalization of Shannon's information theory, that is, to Rényi's information theory, be deriving a new information-theoretic equation and by redefining existing ones in a more logical framework, [98]. The purpose of these information-theoretic equations is to enhance the effectiveness of the mutual information from Shannon's information theory, in finding structural information hidden in the dynamics of models of complex systems and networks. The thesis is based on the following articles: [99], [95], [98] and [100]. The common theme of these articles is the study of structure-dynamics relationships

in models of complex systems and networks, by means of information theory.

In the article [95], we developed an information-theoretic methodology, based on Kolmogorov complexity, to analyse the dynamical output of an executable model of the cytokine regulation in two populations of T cells of the human immune system. We proved that it is a feasible framework to gain biological information from the dynamical behaviour of the system, under different experimental conditions. The methods based on Kolmogorov complexity that we developed can handle the complexity of this executable model, in a straightforward manner. These analysis methods can be implemented easily without much programming difficulty. The predictions and new hypotheses about the behaviour of the model can be easily performed with our methodology. Algorithmic information theory is very well suited to tackle the complexity of the encoding of the output of the executable model. It makes the analyses straightforward, such that the biologists can use the model and our analysis methods, to focus on the conclusions and the predictions provided by them.

We classified different structural knock-out perturbations of the executable model of the complex human immune system, using information extracted from their dynamics. We found biological information from the dynamical trajectories. We made predictions to the amount of efficiency required for a structural knock-out perturbation, to produce a significant change in the overall dynamical behaviour of the system. Such analyses would not have been possible with the model-dependent tools from the GemCell framework of the modelling of reactive systems [5]. In addition, we proved that information theory is a general framework to analyse executable models of complex reactive biological systems. Information theory does not only deal with the reliable transmission of information and with applications in communications engineering. But, it has novel and crucial applications to the study of how information is transmitted and processed in complex systems and networks, and to the discovery of their structure-dynamics relationships.

In the article [99], we implemented an experimental order parameter, based on the normalized compression distance (NCD) from algorithmic information theory, which indicated the dynamical regime of random Boolean networks, from a limited number of dynamical states. The aim of the NCD analysis of the time series of such networks was to determine their dynamical regime, using only the dynamical information form a small number of states. Extensive perturbation experiments to sample the state space of a system are prohibitive in the case of real biological systems. Here, very few measurements could be taken of the state of the system. We simulated the real-life

situation of biological measurements, where a perturbation analysis is not feasible. We constructed an experimental NCD based Derrida curve, as an extension to the one applied to random Boolean networks in [85], and computed an experimental order parameter from it. Using this order parameter, we classified random Boolean networks into different dynamical categories.

In the article [100], we studied the dynamics to structure relationship, in random Boolean networks, in an ensemble approach. The order parameter of [104], termed the expectation of the average sensitivity, represents the theoretical structure to dynamics relationship, showing how to obtain the dynamical class of a random Boolean network, given two key structural parameters. Here, we took the converse approach: starting from the dynamics, we showed that only certain structural classes could give rise to the observed dynamical behaviour. We created a mapping of the dynamical states of random Boolean networks, to structural classes, without inferring the topology. We proved that structural information was present in the dynamics of such networks and that this information could be extracted from the dynamics, without computing any structural or dynamical parameters. We defined the mapping as an algorithm, based on the normalized time-delayed mutual information from Shannon's information theory. The method classified, with a high degree of accuracy, random Boolean networks into different structural classes, using features extracted from their dynamics. We showed that the dynamical behaviour of the network is shaped by its structure. That is, not all possible structures can give rise to all possible dynamics, but that classes of structures are related to classes of dynamics in nontrivial ways. We achieved this by creating the dynamics to structure mapping and applying it to separate random Boolean networks into structural classes. If any structure gave rise to any kind of dynamics, we would not be able to distinguish the structural class of the networks, using only their dynamical behaviour.

In the article [98], we introduced a new information-theoretic equation from Rényi's information theory, which is a generalization of Shannon's theory. We named this equation the partial Rényi transfer entropy. Research carried out in the field of information theory is two-fold: on the one hand, new applications are emerging, outside of the traditional fields of reliable communication, that is, Shannon's information theory, and of the complexity of objects, that is, algorithmic information theory. On the other hand, new information-theoretic equations are derived that can better quantify the structure-dynamics relationship in complex systems and networks. We provided a review of the main equations from Shannon's information

theory and their generalizations from Rényi's theory, which extend the applications of the mutual information in the study of the structure-dynamics relationships in such systems. In addition, we redefined some of the previously introduced equations from Rényi's theory, in a unified framework and we gave the logical reasons supporting these derivations.

The study of structure-dynamics relationships entails defining the space of all possible structures and that of all possible dynamics and possible partitions of these two aspects into classes of structures and classes of dynamics, respectively. Most importantly, the question is how to relate them, through equations or algorithms that would enable the recovery of the information about one part from the information about the other.

The generalizations of the mutual information require a huge computational effort and very powerful estimation methods, to take into account extensive histories of the processes involved, including the environment. This represents one major drawback that hinders the wide use of these methods in the study of the structure-dynamics relationships in complex systems and networks. Together with introducing new multidimensional information-theoretic equations, developing estimation methods for the existing ones is of great importance. These methods are required to handle high dimensions of the random processes involved and to produce very accurate results. A great deal of research effort needs to be carried out in this direction, to make these powerful multidimensional information-theoretic equations feasible in practice, in the study of real-world complex systems and networks.

We have seen that information theory contains a broad range of methods, suitable for diverse applications, which share the same principle: that the complex system or network involved in the application is an information processing system. Conceptually, all the methods from information theory deal with the quantification and transmission of information within complex systems and networks: those pertaining to Shannon's information theory and its generalizations concern the information of a distribution and those pertaining to Kolmogorov complexity concern the information of individual objects. The difference between these two formalisms makes some information-theoretic methods better suited to certain applications, than others. For example, in the case of the executable model, algorithmic information theory provides better results and makes the analysis methods more straightforward to implement, than computing probability distributions for a small subset of symbols that encode some element or action of the system. Probabilistic descriptions of the executable

model would be highly cumbersome and the analysis methods highly problematic. The reason is the complex encoding of the state of the system, as symbols in a text file. The studies are much easier to conduct by compressing the files with real-world compressors, to produce the NCD values and to show how the similarity of states evolves in time. In other cases, such as the random Boolean network model, both frameworks provide suitable results and conclusions. The choice of information-theoretic analysis framework is determined by which description better captures the dynamical behaviour of the system under investigation.

In order to discover knowledge on the structure-dynamics relationships in complex systems and networks, we exploit the fact that these systems are information processing systems. By measuring how information is transmitted and processed in such systems, we can use the paradigm of information theory, to bring further understanding of their organizational principles. We explore how information is processed in these systems, to characterize their structure, their function and the relationship between the two. Information theory was initially devised for communication channels, as a mathematical framework for how to transmit information reliably over error prone channels, in Shannon's information theory, and as a means of quantifying the complexity of objects, in Kolmogorov complexity. However, novel applications of these concepts to the structure-dynamics relationships in complex systems and networks are now emerging. Information theory offers a vast array of methods that can be applied to any type of model class, as long as the model can be described in terms of how it processes and exchanges information. In our work, by analysing the executable model and the random Boolean network model, we brought further evidence that information theory is a universal analysis framework, which is independent of the class of models that is investigated.

Bibliography

[1] Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97.

[2] Aldana, M. (2003). Boolean dynamics of networks with scale-free topology. *Physica D*, 185:45–66.

[3] Aldana, M., Coppersmith, S., and Kadanoff, L. P. (2002). Boolean dynamics with random couplings. *in Perspectives and Problems in Nonlinear Science, Springer*.

[4] Alon, U. (2007). Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8(6).

[5] Amir-Kroll*, H., Sadot*, A., Cohen, I. R., and Harel, D. (2008). GemCell: A generic platform for modeling multi-cellular biological systems. *Theoretical Computer Science*, 391:276–290. *Equal contribution.

[6] Ayer, M., Brunk, H., Ewing, G., Reid, W., and Silverman, E. (1955). An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, 26(4):641–647.

[7] Balleza, E., Alvarez-Buylla, E. R., Chaos, A., Kauffman, S., Shmulevich, I., and Aldana, M. (2008). Critical dynamics in genetic regulatory networks: examples from four kingdoms. *PLoS One*, 3((6):e2456).

[8] Bar-Yam, Y. (1997). *Dynamics of complex systems*. Addison-Wesley.

[9] Bar-Yam, Y. (2003). *Unifying principles in complex systems in "Converging technologies for improving human performance"*. Kluwer, eds. m.c. roco and w.s. bainbridge edition.

[10] Barabási, A.-L. (2005). Taming complexity. *Nature Physics*, 1:68–70.

[11] Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

[12] Baraniuk, R. G., Flandrin, P., Janssen, A. J., and Michel, O. J. (2001). Measuring time-frequency information content using the Rényi entropies. *IEEE Transactions on Information Theory*, 47(4):1391–1409.

[13] Bennett, C. H., Gács, P., Li, M., Vitányi, P. M. B., and Zurek, W. H. (1998). Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423.

[14] Best, M. J. and Chakravarti, N. (1990). Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47:425–439.

[15] Bilke, S. and Sjunnesson, F. (2001). Stability of the Kauffman model. *Physical Review E*, 65:016129–1–5.

[16] Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., and Hwang, D.-U. (2006). Complex networks: structure and dynamics. *Physics Reports*, 424:175–308.

[17] Bollobás, B. and Riordan, O. (2004). The diameter of a scale-free random graph. *Combinatorica*, 24(1):5–34.

[18] Borg, I. and Groenen, P. J. (2005). *Modern Multidimensional Scaling Theory and Applications*. Springer, 2^{nd} edition.

[19] Botev, Z. I., Grotowski, J. F., and Kroese, D. (2010). Kernel density estimation via diffusion. *The Annals of Statistics*, 38(5):2916–2957.

[20] Brenu, E., Staines, D., Tajouri, L., Huth, T., Ashton, K., and Marshall-Gradisnik, S. (2013). Heat shock proteins and regulatory T cells. *Autoimmune Diseases*.

[21] Chen, W., Jin, W., Hardigan, N., Lei, K., Li, L., Marinos, N., McGrady, G., and Wahl, S. M. (2003). Conversion of peripheral $CD4^+CD25^-$ naive T cells to $CD4^+CD25^-$ regulatory T cells by TGF-β induction of transcription factor *Foxp3*. *The Journal of Experimental Medicine*, 198(12):1875–1886.

[22] Cilibrasi, R. and Vitányi, P. M. (2005). Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545.

[23] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20:273–297.

[24] Corthay, A. (2009). How do regulatory T cells work? *Scandinavian Journal of Immunology*, 70(4):326–336.

[25] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory*. John Wiley & Sons, 2^{nd} edition.

[26] Csiszár, I. (1995). Generalized cutoff rates and rényi's information measures. *IEEE Transactions on Information Theory*, 41(1):26–34.

[27] Dai, Y., Han, J., Liu, G., Sun, D., Yin, H., and Yuan, Y.-X. (1999). Convergence properties of nonlinear conjugate gradient method. *SIAM Journal on Optimization*, 10(2):345–358.

[28] de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology*, 9(1):67–103.

[29] de Leeuw, J., Hornik, K., and Mair, P. (2009). Isotone optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and active set methods. *Journal of Statistical Software*, 32(5).

[30] Derrida, B. and Pomeau, Y. (1986). Random network of automata: a simple annealed approximation. *Europhysics Letters*, 1(2):45–49.

[31] Derrida, B. and Stauffer, D. (1986). Phase transitions in two-dimensional Kauffman cellular automata. *Europhysics Letters*, 2(10):739–745.

[32] Drossel, B. (2005). Number of attractors in random Boolean networks. *Physical Review E*, 72(016110):1–5.

[33] Drossel, B., Mihaljev, T., and Greil, F. (2005). Number and length of attractors in a critical Kauffman model with connectivity one. *Physical Review Letters*, 94:088701–1–4.

[34] Erdős, P. and Rényi, A. (1959). On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297.

[35] Erdős, P. and Rényi, A. (1961). On the evolution of random graphs. *Bull. Inst. Internat. Statist.*, 38(4):343–347.

[36] Erdogmus, D., Hild, K. E., Principe, J. C., Lazaro, M., and Santamaria, I. (2004). Adaptive blind deconvolution of linear channels using Rényi's entropy with Parzen window estimation. *IEEE Transactions on Signal Processing*, 52(6):1489–1498.

[37] Erdogmus, D. and Principe, J. C. (2002). Generalized information potential criterion for adaptive system training. *IEEE Transactions on Neural Networks*, 13(5):1035–1044.

[38] Fagiolo, G. (2007). Clustering in complex directed networks. *Physical Review E*, 76:026107–1–8.

[39] Fauré, A., Naldi, A., Chaouiya, C., and Thieffry, D. (2006). Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):e124–e131.

[40] Fisher, J. and Henzinger, T. A. (2007). Executable cell biology. *Nature Biotechnology*, 25(11):1239–1249.

[41] Freeman, L. C. (1977). A set of measures for centrality based on betweenness. *Sociometry*, 40(1):35–41.

[42] Frenzel, S. and Pompe, B. (2007). Partial mutual information for coupling analysis of multivariate time series. *Physical Review Letters*, 99:204101–1–4.

[43] Galas, D. J., Nykter, M., Carter, G. W., Price, N. D., and Shmulevich, I. (2010). Biological information as set-based complexity. *IEEE Transactions on Information Theory*, 56(2).

[44] Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826.

[45] Gokcay, E. and Principe, J. C. (2002). Information theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):158–171.

[46] Goldenfeld, N. and Kadanoff, L. P. (1999). Simple lessons from complexity. *Science*, 284(5411):87–89.

[47] Gray, R. M. and Davisson, L. D. (2010). *An introduction to statistical signal processing*. Cambridge University Press.

[48] Harel, D. (1987). Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274.

[49] Harvey, I. and Bossomaier, T. (1997). Time out of joint: attractors in asynchronous random Boolean networks. In *Proceedings of the Fourth European Conference on Artificial Life (ECAL1997)*, pages 67–75. MIT Press.

[50] Henderson, B. and Pckley, A. G., editors (2005). *Molecular chaperones and cell signaling*. Cambridge University Press.

[51] Hsu, H. P. (1997). *Schaum's outline of Probability, random variables and random processes*. McGraw-Hill.

[52] Ihler, A. and Mandel, M. (2003). Kernel density estimation toolbox for Matlab. *http://www.ics.uci.edu/~ihler/code/kde.html*.

[53] Ivanov, I. and Dougherty, E. R. (2006). Modeling gene regulatory networks: continuous or discrete? *Journal of Biological Systems*, 14(02):219–229.

[54] Janeway, C., Travers, P., Walport, M., and Shlomchik, M. (2001). *Immunobiology: The immune system in health and disease*. Garland Publishing, 5 edition.

[55] Jensen, J. (1906). Sur le fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193.

[56] Jizba, P., Kleinert, H., and Shefaat, M. (2012). Rényi's information transfer between financial time series. *Physica A*, 391(2971-2989).

[57] Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467.

[58] Kaufman, V. and Drossel, B. (2005). On the properties of cycles of simple Boolean networks. *The European Physical Journal B*, 43:115–124.

[59] Klotz, J. G., Kracht, D., Bossert, M., and Schober, S. (2014). Canalizing Boolean functions maximize mutual information. *IEEE Transactions on Information Theory*, 60(4):2139–2147.

[60] Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problemy Peredachi Informatsii*, 1(1):3–11.

[61] Krawitz, P. and Shmulevich, I. (2007a). Basin entropy in Boolean networks ensembles. *Physical Review Letters*, 98:158701–1–4.

[62] Krawitz, P. and Shmulevich, I. (2007b). Entropy of complex relevant components of boolean networks. *Physical Review E*, 76:036115–1–7.

[63] Kruskal, J. (1964a). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27.

[64] Kruskal, J. (1964b). Nonmetric multidimensional scaling: A numerical method. *Psychometrika*, 29(2):115–129.

[65] Kullback, S. and Leibler, R. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

[66] Ladyman, J., Lambert, J., and Wiesner, K. (2013). What is a complex system? *European Journal for Philosophy of Science*, 3(1):33–67.

[67] Lawrence, E. (1995). *Henderson's dictionary of biological terms.* Longman Scientific & Technical, 11 edition.

[68] Li, M., Chen, X., Li, X., Ma, B., and Vitányi, P. M. B. (2004). The similarity metric. *IEEE Transactions on Information Theory*, 50(12):3250–3264.

[69] Li, M. and Vitányi, P. M. B. (2008). *An introduction to Kolmogorov complexity and its applications.* Springer, 3^{rd} edition.

[70] Li, Z. and Srivastava, P. (2004). Heat-shock proteins. *Current Protocols in Immunology*, 58.

[71] Luque, B. and Solé, R. V. (2000). Lyapunov exponents in random Boolean networks. *Physica A*, 284:33–45.

[72] Mäki-Marttunen, T., Kesseli, J., and Nykter, M. (2011). Of the complex of Boolean network state trajectories. In *Proceedings of the* 8^{th} *International Workshop on Computational Systems Biology (WCSB 2011)*, pages 137–140.

[73] Mäki-Marttunen, T., Kesseli, J., and Nykter, M. (2013). Balance between noise and information flow maximizes set complex of network dynamics. *PLoS ONE*, 8(3):e56523.

[74] Malvestuto, F. M. (1986). Statistical treatment of the information content of a database. *Information Systems*, 11(3):211–223.

[75] Miles, R. E. (1959). The complete amalgamation into blocks, by weighted means, of a finite set of real numbers. *Biometrika*, 46(3/4):317–327.

[76] Mitrinović, D. (1970). *Analytic Inequalities*. Springer -Verlag.

[77] Morejon, R. A. and Principe, J. C. (2004). Advanced search algorithms for information-theoretic learning with kernel-based estimators. *IEEE Transactions on Neural Networks*, 15(4):874–884.

[78] Morel, P. A., Faeder, J. R., Hawse, W. F., and Miskov-Zivanov, N. (2014). Modeling the T cell immune response: a fascinating challenge. *Journal of Pharmacokinetics and Pharmacodynamics*, 41(5):401–413.

[79] Neuman, M. (2003). The structure and function of complex networks. *SIAM Review*, 45(2):167–256.

[80] Newman, M. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46(5).

[81] Newman, M., Strogatz, S., and Watts, D. (2001). Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64(026118).

[82] Newman, M. E. J. (2011). Complex systems: A survey. *American Journal of Physics*, 79(800).

[83] Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Reiew E*, 69:026113–1–15.

[84] Nykter, M., Kesseli, J., Shmulevich, I., and Yli-Harja, O. (2006). Analyzing Boolean network dynamics using attractor basin structure. In *Proceedings of the 4^{th} TICSP Workshop on Computational Systems Biology, WCSB 2006*.

[85] Nykter, M., Price, N. D., Aldana, M., Ramsey, S. A., Kauffman, S. A., Hood, L. E., Yli-Harja, O., and Shmulevich, I. (2008a). Gene expression dynamics in the macrophage exhibit criticality. *PNAS*, 105(6):1897–1900.

[86] Nykter, M., Price, N. D., Larjo, A., Aho, T., Kauffman, S. A., Yli-Harja, O., and Shmulevich, I. (2008b). Critical networks exhibit maximal information diversity in structure-dynamics relationships. *Physical Review Letters*, 100:058702–1–4.

[87] Peixoto, T. P. and Drossel, B. (2009). Noise in random Boolean networks. *Physical Review E*, 79:036108–1–9.

[88] Póczos, B. and Schneider, J. (2012). Nonparametric estimation of conditional information and divergences. In *Proceedings of the 2012 International Conference on AI and Statistics (AISTATS 2012)*, volume XX, pages 914–923.

[89] Principe, J. C. (2010). *Information theoretic learning: Rényi's entropy and kernel perspectives*. Springer.

[90] Rached, Z., Alajaji, F., and Campbell, L. L. (2001). Rényi's divergence and entropy rates for finite alphabet Markov sources. *IEEE Transactions on Information Theory*, 47(4):1553–1561.

[91] Raven, P. H. and Johnson, G. B. (1996). *Biology*. WCB/McGraw-Hill.

[92] Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the fourth Berkeley Symposium on Mathematics, Statistics and Probability*, volume 1, pages 547–561.

[93] Ribeiro, A. S. and Kauffman, S. A. (2007). Noisy attractors and ergodic sets in models of gene regulatory networks. *Journal of Theoretical Biology*, 247:743–755.

[94] Ribeiro, A. S., Kauffman, S. A., Lloyd-Price, J., Samuelsson, B., and Socolar, J. E. (2008). Mutual information in random Boolean models of regulatory networks. *Physical Review E*, 77:011901–1–10.

[95] Sadot*, A., Sarbu*, S., Kesseli, J., Amir-Kroll, H., Zhang, W., Nykter, M., and Shmulevich, I. (2013). Information-theoretic analysis of the dynamics of an executable biological model. *PLoS One*, 8(3):e59303. *Equal contribution.

[96] Sakaguchi, S. (2000). Regulatory T cells: Key controllers of immunologic self-tolerance. *Cell*, 101(5):455–458.

[97] Samuelsson, B. and Troein, C. (2003). Superpolynomial growth in the number of attractors in Kauffman networks. *Physical Review Letters*, 90:098701–1–4.

[98] Sarbu, S. (2014). Rényi information transfer: partial Rényi transfer entropy and partial Rényi mutual information. In *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*.

[99] Sarbu, S., Kesseli, J., and Nykter, M. (2012). Quantifying the relationship between the structure and the dynamics of random Boolean networks from time series data. In *Proceedings of the Ninth International Workshop on Computational Systems Biology (WCSB 2012)*.

[100] Sarbu, S., Shmulevich, I., Yli-Harja, O., Nykter, M., and Kesseli, J. (2015). Mapping dynamical states to structural classes for Boolean networks using a classification algorithm. In *Proceedings of the 2015 European Signal Processing Conference (EUSIPCO 2015)*.

[101] Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters*, 85(2):461–464.

[102] Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.

[103] Shmulevich, I., Dougherty, E. R., Kim, S., and Zhang, W. (2002). Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274.

[104] Shmulevich, I. and Kauffman, S. A. (2004). Activities and sensitivities in Boolean network models. *Physical Review Letters*, 93(4):048701–1–4.

[105] Shmulevich, I., Kauffman, S. A., and Aldana, M. (2005). Eukaryotic cells and dynamically ordered or critical but not chaotic. *Proceedings of the National Academy of Sciences of the United States of America*, 102(38):13439–13444.

[106] Solé, R. V. and Luque, B. (1995). Phase transitions and antichaos in generalized Kauffman networks. *Physics Letters A*, 196:331–334.

[107] Strogatz, S. H. (2001). Exploring complex networks. *Nature*, 410:268–276.

[108] Tsan, M.-F. and Gao, B. (2009). Heat shock proteins and immune system. *Journal of Leukocyte Biology*, 85(6):905–910.

[109] Vakorin, V. A., Krakovska, O. A., and McIntosh, A. R. (2009). Confounding effects of indirect connections on causality estimation. *Journal of Neuroscience Methods*, 184:152–160.

[110] van Eeden, C. (1957). A least squares inequality for maximum likelihood estimates of ordered parameters. *Proceedings of the Koninkllijke Nederlandse Akademie van Wetenschappen (A) 60/ Indagtiones Mathematicae 19*, pages 513–521.

[111] Vignali, D. A. (2012). Mechanisms of Treg suppression: still a long way to go. *Frontiers in Immunology*, 3(191).

[112] Wallin, R. P., Lundqvist, A., Moré, S. H., von Bonin, A., Kiessling, R., and Ljunggren, H.-G. (2002). Heat-shock proteins as activators of the innate immune system. *TRENDS in Immunology*, 23(3):130–135.

[113] Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393:440–442.

[114] Zanin-Zhorov, A., Cahalon, L., Tal, G., Margalit, R., Lider, O., and Cohen, I. R. (2006). Heat shock protein 60 enhances $CD4^{+}CD25^{+}$ regulatory T cell function via innate $TLR2$ signaling. *Journal of Clinical Investigation*, 116(7):2022–2032.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3734-9
ISSN 1459-2045